N69-36427
NASA CR-102233

DYNAMIC STABILITY OF CYLINDRICAL PROPELLANT TANKS

by

Daniel D. Kana
Wen-Hwa Chu

FINAL REPORT, PART II
Contract No. NAS8-21282
Control No. DCN 1-8-75-00009 (1F)

SwRI Project No. 02-2332

Prepared for

National Aeronautics and Space Administration
George C. Marshall Space Flight Center
Huntsville, Alabama

2 June 1969

SOUTHWEST RESEARCH INSTITUTE
SAN ANTONIO HOUSTON

SOUTHWEST RESEARCH INSTITUTE
Post Office Drawer 28510, 8500 Culebra Road
San Antonio, Texas 78228

DYNAMIC STABILITY OF CYLINDRICAL PROPELLANT TANKS

by

Daniel D. Kana
Wen-Hwa Chu

FINAL REPORT, PART II

Contract No. NAS8-21282
Control No. DCN 1-8-75-00009 (IF)

SwRI Project No. 02-2332

Prepared for

National Aeronautics and Space Administration
George C. Marshall Space Flight Center
Huntsville, Alabama

2 June 1969

Approved:

H. Norman Abramson, Director
Department of Mechanical Sciences

PREFACE

This report constitutes the second of two volumes which summarize the work accomplished under Contract NAS8-21282. It contains supporting experimental data, a theoretical analysis, and a listing of a digital computer program designed for predicting dynamic stability of propellant tanks under longitudinal excitation. The first part of the work, which deals with the dynamic state of the system prior to instability, is summarized in Final Report, Part I, entitled "Influence of a Rigid Top Mass on the Response of a Pressurized Cylinder Containing Liquid."

Both Part I and Part II of this Final Report are published on the same date. They present significant extensions and refinements of concepts originated under previous investigations conducted for NASA-MSFC. Results of this preliminary work are summarized in "Dynamic Stability and Parametric Resonance in Cylindrical Propellant Tanks," by Daniel D. Kana, Wen-Hwa Chu, and Tom D. Dunham, Final Report, Contract No. NAS8-20329, SwRI Project No. 02-1876, January 17, 1968.

ABSTRACT

Dynamic instability and associated parametric resonance is a dominant form of response in a longitudinally excited cylindrical shell containing liquid. In order to assess the significance of such responses in a space vehicle propellant tank, the present paper is devoted to a theoretical and experimental study of their occurrence in a cylindrical shell system which includes the influences of axial preload, ullage pressure, partial liquid depth, and a finite top impedance. Donnell shell theory along with a modified Galerkin procedure is utilized to formulate equations which govern the stability of perturbations superimposed on an axisymmetric initial state of response. Stability boundaries are computed for a range of parameters affecting the region of principal parametric resonance and are compared with experimental results. It is found that liquid depth, top impedance, and ullage pressure have a strong influence on stability, while the effects of axial preload are relatively insignificant.

TABLE OF CONTENTS

NOMENCLATURE

a	radius of the shell
c_0	speed of sound in the liquid
c_s	E/ρ_s, speed of stress waves in the shell
E	modulus of elasticity
g	standard acceleration of gravity
H	h/a, nondimensional liquid depth
H_s	h_s/a, nondimensional thickness of shell
I_z	mass moment of inertia of top weight about z axis
ℓ	length of the shell
m	one-half of the number of circumferential nodes; $\cos(m\theta)$
$N^*_{xxd}, N^*_{\theta\theta d}$	dynamic part of initial-state stress resultants [nondimensionalized by $(1 - \nu^2)/Eh_s$]
$N^*_{xxs}, N^*_{\theta\theta s}, N^*_{x\theta s}$	static part of initial-state stress resultants [nondimensionalized by $(1 - \nu^2)/Eh_s$]
n	axial wave number; $\sin n\pi x/\ell$
P_r	nondimensional pressure loading on shell, P_r/E
P_0, p_0	axial preload, ullage pressure
R, θ, X	cylindrical coordinates (space-fixed) nondimensionalized by radius a
U, V, W	shell displacements u, v, w, nondimensionalized by the radius a
X_0	nondimensional amplitude of axial excitation ($X_0 = \hat{x}_0/a$)
Z_0	top acceleration impedance (force/acceleration)
β	density parameter $\rho_\ell a/\rho_s h_s$

NOMENCLATURE (Cont'd)

ν	Poissons ratio
Φ	velocity potential, nondimensionalized by $\omega_0^2 a^2/\omega_r$
ρ_ℓ	mass density of liquid
ρ_s	mass density of the shell
τ	nondimensional time, $\tau = \omega_r t$
ω_0^2	liquid parameter c_0^2/a^2
ω_r	response frequency
ω	excitation frequency
ω_k	natural frequency of m-k'th mode
Ω_i^2	designated frequency, nondimensionalized by a^2/c_s^2
$\tilde{\Omega}_i^2$	designated frequency, nondimensionalized by $(1 - \nu^2) \times a^2/c_s^2$
$\underset{\sim}{\Omega}_i^2$	designated frequency, nondimensionalized by a^2/c_0^2

Superscripts

$(\hat{\ })$	the amplitude of ( )
$(\dot{\ })$	$(d/d\tau)$ ( ), $\tau = \omega t$
$(\ )^p$	related to initial state response

LIST OF ILLUSTRATIONS

INTRODUCTION

Dynamic instability and parametric resonance are known to occur in many engineering systems. Numerous classic examples have been studied in detail by Bolotin[1]. In more recent experimental investigations[2], it was found that this type of behavior is dominant amidst a complex variety of responses which can be observed in a longitudinally excited model vehicle propellant tank which is not sufficiently reinforced with stiffeners. A theoretical and further experimental investigation[3] was conducted for a longitudinally excited, liquid-filled cylindrical shell. It was found that the system initially tends to respond in a state comprised of linear axisymmetric modes. However, the resulting membrane stresses form a parametric load with respect to nonaxisymmetric perturbations superimposed on the initial state. Thus, for wide ranges of the excitation parameters, instability and subsequent parametric resonance results, and linear vibration theory is no longer adequate to predict the response of either liquid pressure or wall motion.

In order to assess the significance of such instabilities in a propellant tank which forms a component in an overall space vehicle structure, the present paper is devoted to a study of their occurrence in a cylindrical shell system which includes the influences of axial preload, ullage pressure, partial liquid depth, and a finite top impedance. A diagram of the system is shown in Figure 1, which includes the appropriate parameters and boundary conditions for both the initial and perturbed states. The initial state represents linear forced axisymmetric motion, whose responses have already

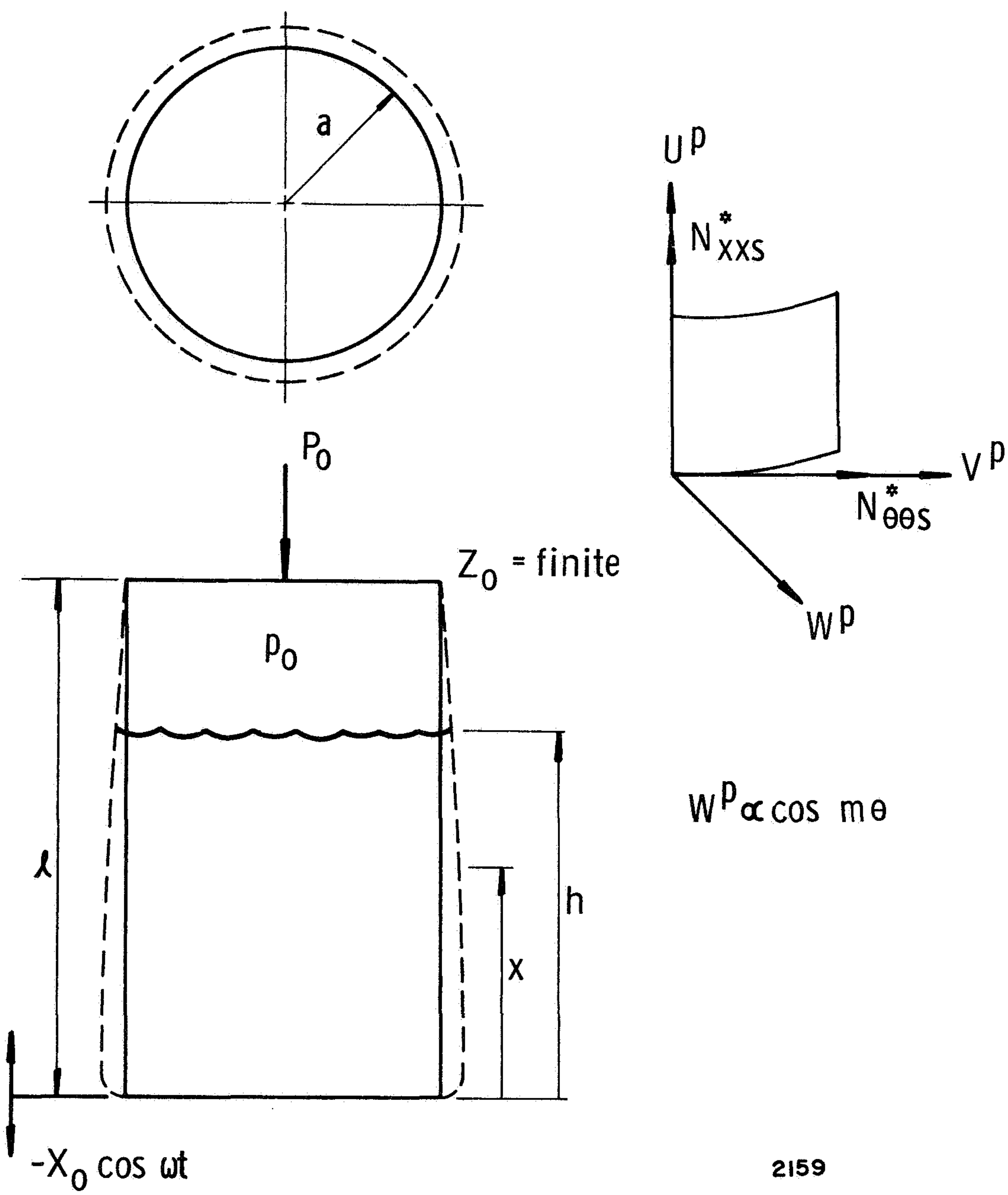


Figure 1a. Mechanism Of Dynamic Instability - Initial State $m = 0$

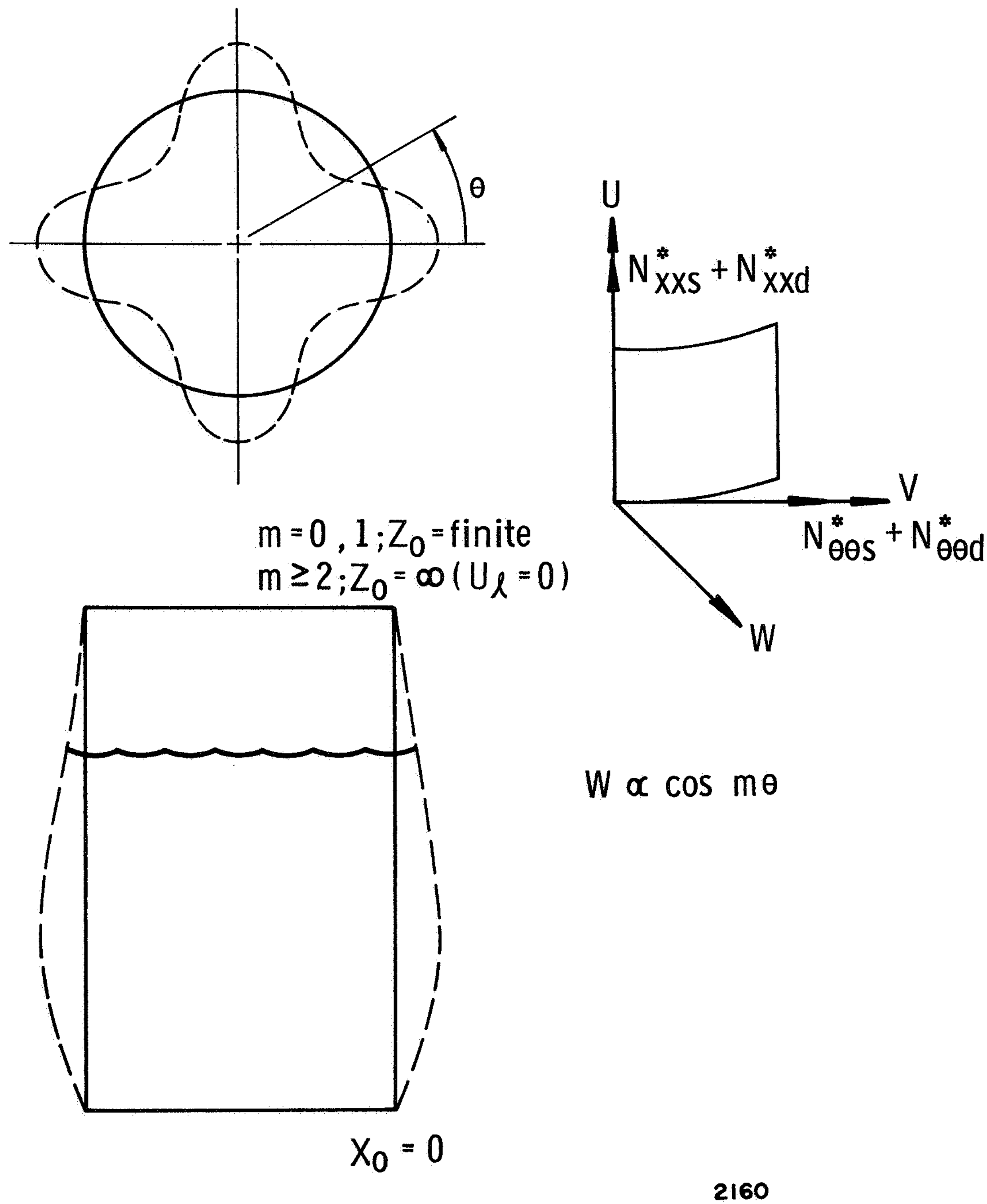


Figure 1b. Mechanism Of Dynamic Instability – Perturbed State
m ≥ 0

been determined, along with natural frequencies and modal functions for the system[4]. Stability of motion in the perturbed state is the subject of the present paper, although results from Reference 4 must be utilized in part of the analysis. Note that, theoretically, the perturbed state can be either axisymmetric ($m = 0$) or nonaxisymmetric ($m > 0$); however, for a single tank system, the nonaxisymmetric form of instability is dominant.

DERIVATION OF STABILITY EQUATIONS

The perturbed motion represented by Figure 1b will be analyzed by means of Sander's nonlinear shell equations[5,6] which are based on Donnell approximations. These equations contain nonlinear terms resulting from the rotation of shell elements as well as nonlinear strain-displacement relations. We will follow the philosophy of Bolotin[1] and assume that retaining only nonlinear terms which result from rotations is sufficient to determine dynamic stability. Compressible flow theory is used for the liquid. The motion is expanded into a series of the natural mode eigenvectors which were obtained from the solution of the free vibration problem[4]. A modified Galerkin procedure is then utilized to reduce the system to a linear second-order, time dependent set of coupled differential equations having periodic coefficients. The method is "modified" in the sense that the natural modal functions (finite series eigenvectors) are chosen as weighting functions, although they are not of closed form. An approximation of the perturbed motion will then be obtained by the use of only one eigenvector term of the series, so that the coupled set reduces to a single stability equation.

Thus, the governing shell equations are of the form

$$F_1 = L_{11}U + L_{12}V + L_{13}W - \tilde{\Omega}_r^2 \partial^2 U/\partial\tau^2 = 0 \tag{1a}$$

$$F_2 = L_{21}U + L_{22}V + L_{23}W - \tilde{\Omega}_r^2 \partial^2 V/\partial\tau^2 = 0 \tag{1b}$$

$$F_3 = L_{31}U + L_{32}V + L_{33}W - \tilde{\Omega}_r^2 \partial^2 W/\partial\tau^2 + \epsilon_d \tilde{L}_{33} W \cos \omega t + \frac{(1 - \nu^2)}{H_s} P_r = 0 \tag{1c}$$

where

$$L_{11} = \frac{\partial^2}{\partial X^2} + \frac{1 - \nu}{2} \frac{\partial^2}{\partial \theta^2}, \qquad L_{12} = \frac{1 + \nu}{2} \frac{\partial^2}{\partial X \partial \theta}$$

$$L_{13} = \nu \frac{\partial}{\partial X}, \qquad L_{21} = \frac{1 + \nu}{2} \frac{\partial^2}{\partial X \partial \theta}$$

$$L_{22} = \frac{\partial^2}{\partial \theta^2} + \frac{1 - \nu}{2} \frac{\partial^2}{\partial X^2}, \qquad L_{23} = \frac{\partial}{\partial \theta}$$

$$L_{31} = - \nu \frac{\partial}{\partial X}, \qquad L_{32} = - \frac{\partial}{\partial \theta} \tag{2a}$$

$$L_{33} = - \left[\frac{H_s^2}{12}\left(\frac{\partial^4}{\partial X^4} + 2 \frac{\partial^4}{\partial X^2 \partial \theta^2} + \frac{\partial^4}{\partial \theta^4}\right) + 1\right] + \left[\frac{\partial}{\partial X}\left(N^*_{xxs} \frac{\partial}{\partial X}\right) + \frac{\partial}{\partial X}\left(N^*_{x\theta s} \frac{\partial}{\partial \theta}\right) + \frac{\partial}{\partial \theta}\left(N^*_{\theta xs} \frac{\partial}{\partial X}\right) + \frac{\partial}{\partial \theta}\left(N^*_{\theta\theta s} \frac{\partial}{\partial \theta}\right)\right]$$

$$\tilde{L}_{33} = \frac{\partial}{\partial X}\left(\hat{N}^*_{xxd} \frac{\partial}{\partial X}\right) + \frac{\partial}{\partial \theta}\left(\hat{N}^*_{\theta\theta d} \frac{\partial}{\partial \theta}\right) \tag{2b}$$

$\epsilon_d = 0$ for free vibration

$\epsilon_d = 1$ for forced vibration

Pressure loading on the shell is given by:

$$\frac{(1-\nu^2)}{H_s} P_r = -\tilde{\Omega}_0^2 \beta \frac{\partial \Phi}{\partial \tau} \text{ at } R = 1 \tag{3}$$

where the fluid velocity potential is governed by

$$\nabla^{*2}\Phi - \underset{\sim}{\Omega}_r^2 \partial^2\Phi/\partial\tau^2 = 0 \tag{4}$$

Boundary conditions on the fluid and shell are:

At $X = 0$,

$$U = 0, \quad W = 0, \quad V = 0, \quad \partial^2 W/\partial X^2 = 0$$

At $X = \ell/a$,

$$W = 0, \quad V = 0, \quad \partial^2 W/\partial X^2 = 0$$

and

$$F_4 = \partial U/\partial X + Z^{**}\tilde{\Omega}_r^2 \partial^2 U/\partial\tau^2 = 0 \tag{5}$$

where

$$Z^{**} = \frac{Z_0(1-\nu^2)}{2\pi\rho_s a^2 h_s} \quad \text{for } m = 0$$

$$Z^{**} = \frac{(1-\nu^2)I_z}{4\rho_s h_s a^5} \quad \text{for } m = 1$$

$$Z^{**} = \infty \quad \text{for } m \geq 2$$

Solutions of the shell motion having a given circumferential displacement distribution will be sought as expansions of the m, k-th natural modes

$$U(\theta, \tau, X) = \cos m\theta \sum_{k=1}^{K} a_k(\tau) U_{mk}(X) = \sum_{k=1}^{K} a_k U_{1k} \tag{6a}$$

$$V(\theta, \tau, X) = \sin m\theta \sum_{k=1}^{K} a_k(\tau) V_{mk}(X) = \sum_{k=1}^{K} a_k U_{2k} \tag{6b}$$

$$W(\theta, \tau, X) = \cos m\theta \sum_{k=1}^{K} a_k(\tau) W_{mk}(X) = \sum_{k=1}^{K} a_k U_{3k} \tag{6c}$$

where for convenience we have defined a general shell displacement vector

$$\vec{U} = \vec{U}(U, V, W) = \vec{U}(U_1, U_2, U_3)$$

which is a function of both space and time and is associated with the fluid velocity potential Φ and upper shell displacement U_ℓ.

The potential Φ satisfies Equation (4) which forms a constraint on the shell system. In order to interpret the fluid pressure loading as an apparent mass which is valid at the response frequency ω_r, we express the potential for forced motion as

$$\Phi(\theta, \tau, \omega_r, R, X) = \cos m\theta \sum_{k=1}^{K} \dot{a}_k(\tau) \Phi_{mk}(\omega_r, R, X) \tag{7}$$

Note that $\Phi_{mk}(\omega_r, R, X)$ is the component of Φ associated with a shell displacement component W_{mk}, and both liquid and shell motion is anticipated to be nearly periodic with responses at frequency ω_r. At the shell wall, we use the notation

$$\Phi_{mk}(\omega_r, 1, X) = \Phi_{mk}(\omega_r, X)$$

For the special case of $\omega_r = \omega_k$, the system responds in the m, k-th natural mode, and the shell displacement modal functions form the vector

$$\vec{U}_k = \vec{U}_k(U_{1k}, U_{2k}, U_{3k})$$

This vector is a function of space only and is associated with the fluid velocity potential $\Phi_{mk}(\omega_k, X)$ and the top displacement $U_{\ell k}$. From the definition of natural frequencies, these modal functions satisfy

$$\sum_{j=1}^{3} L_{ij}U_{jk} + \delta_{i3}\tilde{\Omega}_0^2\beta\Phi_{mk}(\omega_k, X) + \tilde{\Omega}_k^2 U_{ik} = 0 \qquad i = 1,2,3 \tag{8a}$$

$$\partial U_{1k}/\partial X = Z^{**}\tilde{\Omega}_k^2 U_{\ell k} \text{ at } X = \ell / a \tag{8b}$$

We now consider the forced motion. By means of a Galerkin procedure[7], we form an expression for virtual work in the system

$$\sum_{i=1}^{3} \int_S F_i(\vec{U}) \cdot U_{ik'}\, dS + \epsilon_m \int_C F_4(U_{\ell k}) \cdot U_{\ell k'}\, d\theta = 0 \tag{9}$$

where

$\epsilon_m = 1$ for $m = 0, 1$

$\epsilon_m = 0$ for $m \geq 2$

More specifically, we substitute Equations (3), (6) and (7) into Equations (1) and (5) and then by means of Equation (9) form an expression for virtual work between forces (expressed in terms of displacements) associated with the general forced motion and displacements associated with the m, k'-th natural mode. There results:

$$\sum_{k=1}^{K} \Bigg[a_k \int_S \sum_{i=1}^{3} \sum_{j=1}^{3} U_{ik'}(L_{ij}U_{jk})\, dS$$

$$- \ddot{a}_k \int_S \tilde{\Omega}_0^2 \beta \Phi_{mk}(\omega_r, X) U_{3k'} \cos m\theta \, dS - \tilde{\Omega}_r^2 \ddot{a}_k \int_S \sum_{i=1}^{3} U_{ik'} U_{ik}\, dS$$

$$+ \epsilon_d a_k \cos \omega t \int_S U_{3k'}(\tilde{L}_{33} U_{3k})\, dS + \epsilon_m \int_C \Bigg(a_k \frac{\partial U_{1k}}{\partial X} U_{\ell k'}$$

$$+ Z^{**} \tilde{\Omega}_r^2 \ddot{a}_k U_{\ell k} U_{\ell k'} \Bigg)\, d\theta \Bigg] = 0 \qquad (10)$$

Upon use of Equations (8), reverting back to the more conventional displacement symbols in Equations (6) and carrying out the spatial integration, this can be written as

$$\sum_{k=1}^{K} [(K_{2k'k} - \epsilon_m K_{4k'k})(\tilde{\Omega}_r^2 \ddot{a}_k + \tilde{\Omega}_k^2 a_k) + K_{3k'k}(\tilde{\Omega}_r^2 \ddot{a}_k + \tilde{\Omega}_k^2 a_k)]$$

$$+ \sum_{k=1}^{K} \Big\{ a_k \tilde{\Omega}_k^2 [M_{k'k}(\omega_k) + I_{k'k}] + \ddot{a}_k \tilde{\Omega}_r^2 [M_{k'k}(\omega_r) + I_{k'k}]$$

$$+ a_k N_{k'k} \cos \omega t \Big\} = 0 \qquad (11)$$

where k' = 1, 2, 3, . . . , K, and

$$K_{2k'k} = \int_0^{\ell/a} U_{mk'} U_{mk}\, dX \qquad (12a)$$

$$K_{3k'k} = \int_0^{\ell/a} V_{mk'} V_{mk}\, dX \qquad (12b)$$

$$K_{4k'k} = Z^{**} \int_0^{2\pi} U_{\ell k'} U_{\ell k}\, d\theta \qquad (12c)$$

$$M_{k'k}(\omega_r) = \beta \frac{\omega_0^2}{\omega_r^2} \int_0^{\ell/a} W_{mk'} \Phi_{mk}(\omega_r, X)\, dX \tag{12d}$$

$$M_{k'k}(\omega_k) = \beta \frac{\omega_0^2}{\omega_k^2} \int_0^{\ell/a} W_{mk'} \Phi_{mk}(\omega_k, X)\, dX \tag{12e}$$

$$I_{k'k} = \int_0^{\ell/a} W_{mk'} W_{mk}\, dX \tag{12f}$$

$$N_{k'k} = - \int_0^{\ell/a} W_{mk'} \left(\hat{N}^*_{xxd} \frac{\partial^2 W_{mk}}{\partial X^2} + \frac{\partial \hat{N}^*_{xxd}}{\partial X} \frac{\partial W_{mk}}{\partial X} - m^2 \hat{N}^*_{\theta\theta d} W_{mk} \right) dX \tag{12g}$$

The coupled set of K equations (11) govern the perturbed motion, described in Figure 1b, for a given value of m. We will limit further discussion to the case of modes having $m \geq 2$. For these modes, the dominant motion is radial for the set of natural modes at lower frequencies. Thus, the terms under the first summation can be neglected, and, in matrix notation, Equations (11) become

$$\tilde{\Omega}_r^2 [\underset{\sim}{M}] \left\{ \ddot{a} \right\} + \tilde{\Omega}_k^2 [\underset{\sim}{K}] \left\{ a \right\} + [\underset{\sim}{T}] \left\{ a \right\} \cos \omega t = 0 \tag{13a}$$

where the elements of the k'-th row and k-th column of the corresponding matrices are

$$\underset{\sim}{M}_{k'k} = M_{k'k}(\omega_r) + I_{k'k}, \qquad \underset{\sim}{K}_{k'k} = M_{k'k}(\omega_k) + I_{k'k}$$

$$\underset{\sim}{T}_{k'k} = N_{k'k}; \qquad \text{and} \left\{a\right\} = \begin{Bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ \cdot \\ a_K \end{Bmatrix}$$

Equation (13a) can further be written as

$$\left\{\ddot{a}\right\} + \frac{\tilde{\Omega}_k^2}{\tilde{\Omega}_r^2} [\underset{\sim}{M}]^{-1} [\underset{\sim}{K}] \left\{a\right\} + [\underset{\sim}{M}]^{-1} [\underset{\sim}{T}] \left\{a\right\} \cos \omega t = 0 \tag{13b}$$

When the flow is incompressible, $M_{k'k}$ is independent of ω and we have

$$[\underset{\sim}{M}]^{-1} [\underset{\sim}{K}] = [I]$$

where [I] is the identity matrix. It must be emphasized that Equations (13) are not general differential equations in time but include the restriction of nearly periodic motion in the generalized apparent mass given by Equation (7).

EVALUATION OF MATRIX ELEMENTS

Modal Functions

Elements of the matrices in Equations (13) will now be evaluated from Equations (12d-g) in terms of the X-dependent natural modal functions (eigenvectors) of the system. These functions, which are not of closed form, have previously been determined from an eigenvalue problem[4] in terms of the following series forms for the shell displacements:

$$U_{mk}(X) = \frac{1}{2} B_{2k}X^2 + B_{1k}(X - \ell/a) + B_{m0k}$$

$$+ \sum_{n=1}^{N} B_{mnk} \cos \lambda_n X \qquad (14a)$$

$$V_{mk}(X) = \sum_{n=1}^{N} C_{mnk} \sin \lambda_n X \qquad (14b)$$

$$W_{mk}(X) = \sum_{n=1}^{N} A_{mnk} \sin \lambda_n X \qquad (14c)$$

and, for the velocity potential,

$$\Phi_{mk}(\omega_r, X) = \sum_{n=1}^{N} A_{mnk}\Psi_{mn}(\omega_r, X) \qquad (15)$$

where $\Psi_{mn}(\omega_r, X)$ is a component function which satisfies Equation (4) for vibration at frequency ω_r, and corresponds to the sin $\lambda_n X$ component function in W_{mk} through the boundary condition which must be satisfied at the tank wall[4].

Mass Coefficients

The mass coefficient $I_{k'k}$ will now be developed from Equation (12f). By means of Equation (14c), there results

$$I_{k'k} = \int_0^{\ell/a} \sum_{n=1}^{N} \sum_{n'=1}^{N} A_{mnk}A_{mn'k'} \sin \lambda_{n'}X \sin \lambda_n X \, dX$$

$$= \frac{\ell}{2a} \sum_{n=1}^{N} \sum_{n'=1}^{N} A_{mnk}A_{mn'k'} \delta_{n'n} = \frac{\ell}{2a} \sum_{n=1}^{N} A_{mnk}A_{mnk'} \qquad (16)$$

By substituting Equations (14c) and (15) into (12d), the liquid apparent mass coefficient corresponding to the response frequency ω_r becomes

$$M_{k'k}(\omega_r) = \sum_{n=1}^{N} \sum_{n'=1}^{N} A_{mnk} A_{mn'k'} \mathcal{M}_{n'n}(\omega_r) \tag{17}$$

where

$$\mathcal{M}_{n'n}(\omega_r) = \beta \frac{\omega_0^2}{\omega_r^2} \int_0^{\ell/a} \Psi_{mn}(\omega_r, X) \sin \lambda_{n'} X \, dX$$

Except for a normalizing constant, the latter expression has also been evaluated in previous work. That is,

$$\mathcal{M}_{n'n}(\omega_r) = \alpha_{n'}^2 M_{mn'n} \tag{18}$$

where

$$\alpha_{n'}^2 = \int_0^{\ell/a} \sin^2 \lambda_{n'} X \, dX = \frac{\ell}{2a}$$

and $M_{mn'n}$ is given by Equation (18a) in Reference 4. Note, however, that the free index k used in the referenced expression is not the same k which is used to designate the natural mode herein, and we must also use $\omega_r = \omega$.

Finally, the apparent mass coefficient $M_{k'k}(\omega_k)$ given by Equation (12e) is obtained simply by substituting $\omega_r = \omega_k$ in Equations (17) and (18).

Parametric Coefficients

Upon substitution of Equation (14c) into (12g), we obtain:

$$N_{k'k} = \sum_{n=1}^{N} \sum_{n'=1}^{N} A_{mnk} A_{mn'k'} J_{n'n} \tag{19a}$$

where

$$J_{n'n} = \int_0^{\ell/a} \left(\lambda_n^2 \hat{N}^*_{xxd} \sin \lambda_n X - \lambda_n \frac{\partial \hat{N}^*_{xxd}}{\partial X} \cos \lambda_n X \right.$$
$$\left. + m^2 \hat{N}^*_{\theta\theta d} \sin \lambda_n X \right) \sin \lambda_{n'} X \, dX \tag{19b}$$

The dynamic stress resultant amplitudes $\hat{N}^*_{xxd}$ and $\hat{N}^*_{\theta\theta d}$ are produced by forced excitation in the axisymmetric initial state described in Figure 1a. These stress amplitudes can be expressed in terms of the amplitudes of the initial-state displacements by means of the usual stress-displacement equations

$$\hat{N}^*_{xxd} = \frac{\partial \hat{U}^p}{\partial X} + \nu \hat{W}^p \tag{20a}$$

$$\hat{N}^*_{\theta\theta d} = \hat{W}^p + \nu \frac{\partial \hat{U}^p}{\partial X} \tag{20b}$$

Thus, the parametric coefficients are partly determined by the initial state displacement amplitudes $\hat{U}^p$ and $\hat{W}^p$.

The solution to the linear forced axisymmetric response of the initial state, in terms of displacements, has previously been given by Equation (25) in Reference 4. However, in the direct use of this equation, the appropriate elements of its matrices must include the substitution

$$M^{**} = Z^{**} \text{ for } m = 0, 1 \tag{21}$$

since an arbitrary acceleration impedance is allowed in the present problem, rather than only a rigid mass. Further, to allow for comparison of numerical and experimental data, it is convenient to express the initial-state displacements as ratios of the excitation amplitude X_0. Therefore, the dynamic displacement amplitudes are of the form

$$\hat{U}^p = X_0\left[\frac{1}{2} B_2^p X^2 + B_1^p\left(X - \frac{\ell}{a}\right) + B_{m0}^p + \sum_{n''=1}^{N} B_{mn''}^p \cos \lambda_{n''} X\right] \tag{22a}$$

$$\hat{W}^p = X_0 \sum_{n''=1}^{N} A_{mn''}^p \sin \lambda_{n''} X \tag{22b}$$

whose coefficients are completely determined by solving for the case of $X_0 = 1$.

The initial-state stresses can now be determined. Upon substituting Equations (22) into (20a), there results

$$\hat{N}_{xxd}^* = X_0\left[B_1^p + B_2^p X - \sum_{n''=1}^{N} (\lambda_{n''} B_{mn''}^p - \nu A_{mn''}^p) \sin \lambda_{n''} X\right] \tag{23a}$$

and the derivative is

$$\frac{\partial \hat{N}_{xxd}^*}{\partial X} = X_0\left[B_2^p - \lambda_{n''} \sum_{n''=1}^{N} (\lambda_{n''} B_{mn''}^p - \nu A_{mn''}^p) \cos \lambda_{n''} X\right] \tag{23b}$$

Upon substituting Equations (22) into (20b), there results:

$$\hat{N}^{*}_{\theta\theta d} = X_0 \left[\nu(B_1^p + B_2^p X) + \sum_{n''=1}^{N} (A^p_{mn''} - \nu\lambda_{n''}B^P_{mn''}) \sin \lambda_{n''}X \right] \quad (23c)$$

For convenience of computation, these stress resultants and derivatives are expanded into complete Fourier series as follows:

$$\hat{N}^{*}_{xxd} = X_0 \sum_{n''=1}^{N} N_{1n''} \sin \lambda_{n''}X \quad (24a)$$

$$\hat{N}^{*}_{\theta\theta d} = X_0 \sum_{n''=1}^{N} N_{2n''} \sin \lambda_{n''}X \quad (24b)$$

$$\frac{\partial \hat{N}^{*}_{xxd}}{\partial X} = X_0 \left(B_2^p + \sum_{n''=1}^{N} N_{3n''} \cos \lambda_{n''}X \right) \quad (24c)$$

where

$$N_{1n''} = B_2^p \tilde{X}_{1n''} + B_1^p \tilde{X}_{0n''} - \lambda_{n''}B^p_{mn''} + \nu A^p_{mn''}$$

$$N_{2n''} = A^p_{mn''} + \nu(B_2^p \tilde{X}_{1n''} + B_1^p \tilde{X}_{0n''} - \lambda_{n''}B^p_{mn''})$$

$$N_{3n''} = - \lambda^2_{n''}B^p_{mn''} + \nu\lambda_{n''}A^p_{mn''}$$

and

$$\tilde{X}_{0n''} = \frac{2a}{\ell} \int_0^{\ell/a} \sin \lambda_{n''}X \, dX$$

$$\tilde{X}_{1n''} = \frac{2a}{\ell} \int_0^{\ell/a} X \sin \lambda_{n''}X \, dX$$

The parametric coefficients can now be completely evaluated. Upon substitution of Equations (24) into Equations (19), there results

$$N_{k'k} = X_0 \sum_{n=1}^{N} \sum_{n'=1}^{N} A_{mnk}A_{mn'k'} \left[\sum_{n''=1}^{N} (\lambda_n^2 N_{1n''} d_{n''n'n} + m^2 N_{2n''} d_{n''n'n} - \lambda_n N_{3n''} e_{n''n'n}) - \lambda_n B_2^p e_{0n'n} \right] \tag{25}$$

where

$$e_{0n'n} = \int_0^{\ell/a} \cos \lambda_n X \sin \lambda_{n'} X \, dX$$

$$e_{n''n'n} = \int_0^{\ell/a} \cos \lambda_{n''} X \cos \lambda_n X \sin \lambda_{n'} X \, dX$$

$$d_{n''n'n} = \int_0^{\ell/a} \sin \lambda_{n''} X \sin \lambda_{n'} X \sin \lambda_n X \, dX$$

<u>One-Term Approximation</u>

For the m-k'th natural mode in Equation (13a), set k' = k to obtain

$$\tilde{\Omega}_r^2 \overline{M} \ddot{a}_k + \tilde{\Omega}_k^2 \overline{K} a_k + X_0 \overline{T} a_k \cos \omega t = 0 \tag{26}$$

where from Equations (16-19)

$$\overline{M} = \frac{\ell}{2a} \sum_{n=1}^{N} \sum_{n'=1}^{N} A_{mnk}A_{mn'k}M_{mn'n}(\omega_r) + \frac{\ell}{2a} \sum_{n=1}^{N} A_{mnk}^2$$

$$\overline{K} = \frac{\ell}{2a} \sum_{n=1}^{N} \sum_{n'=1}^{N} A_{mnk}A_{mn'k}M_{mn'n}(\omega_k) + \frac{\ell}{2a} \sum_{n=1}^{N} A_{mnk}^2$$

$$\overline{T} = \sum_{n=1}^{N} \sum_{n'=1}^{N} A_{mnk}A_{mn'k} \left[\sum_{n''=1}^{N} (\lambda_n^2 N_{1n''}d_{n''n'n} + m^2 N_{2n''}d_{n''n'n} - \lambda_n N_{3n''}e_{n''n'n}) - \lambda_n B_2^p e_{0n'n} \right]$$

Equation (26) is a Mathieu equation whose stability properties are well known. To put it in a standard form[8] for determining the stability boundaries for principal parametric (1/2-subharmonic) resonance, we set $\omega = 2\omega_r$ and obtain

$$\ddot{a}_k + (\overline{a} + 2\overline{q}X_0 \cos 2\tau)a_k = 0 \tag{27}$$

where

$$\overline{a} = \frac{\tilde{\Omega}_k^2\overline{K}}{\tilde{\Omega}_r^2\overline{M}}, \qquad \overline{q} = \frac{\overline{T}}{2\tilde{\Omega}_r^2\overline{M}}$$

The stability boundaries can then be approximated by

$$\begin{aligned} \overline{q}X_0 &= \overline{a} - 1 \text{ for } \overline{a} > 1 \\ \overline{q}X_0 &= 1 - \overline{a} \text{ for } \overline{a} < 1 \end{aligned} \tag{28}$$

In terms of input acceleration, which is convenient for experimental measurement, these become

$$g_x = X_0 \frac{\omega^2 a}{g} = \frac{\omega^2 a}{g} \left(\frac{|\overline{a} - 1|}{\overline{q}} \right) \tag{29}$$

THEORETICAL AND EXPERIMENTAL RESULTS

Experimental data for stability boundaries are obtained from the apparatus shown in Figure 2. All pertinent parameters, including the input

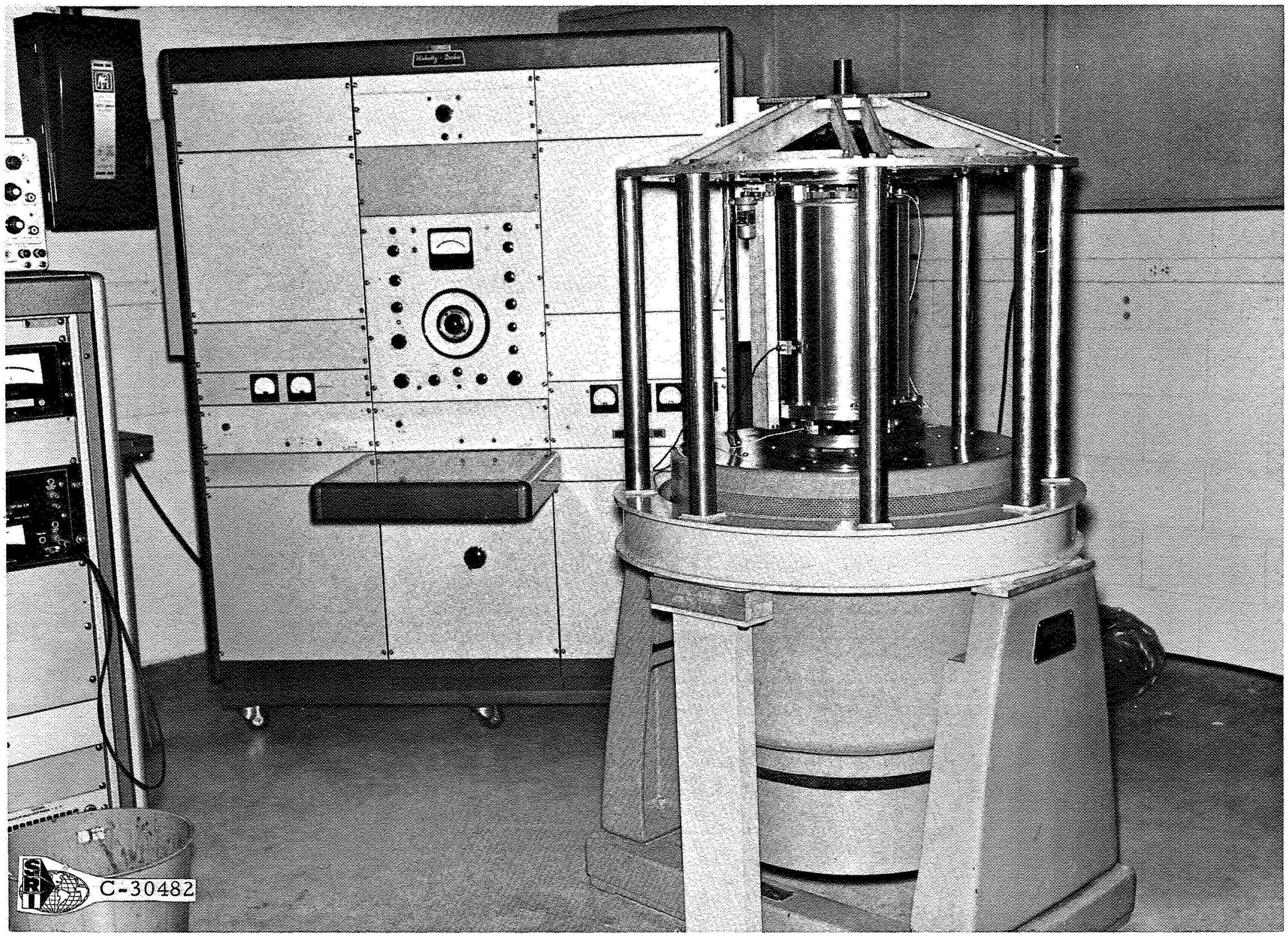
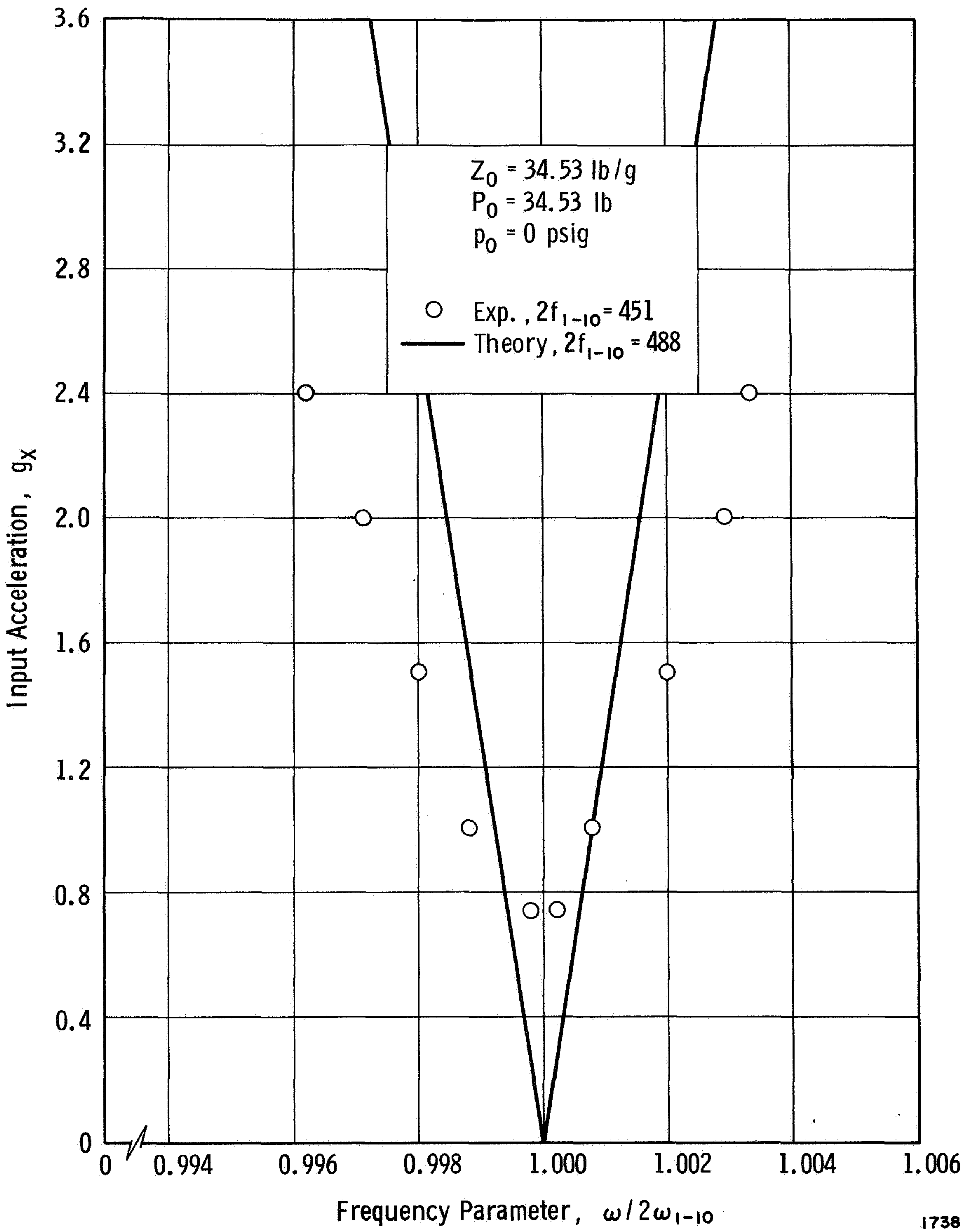


Figure 2. Experimental Apparatus

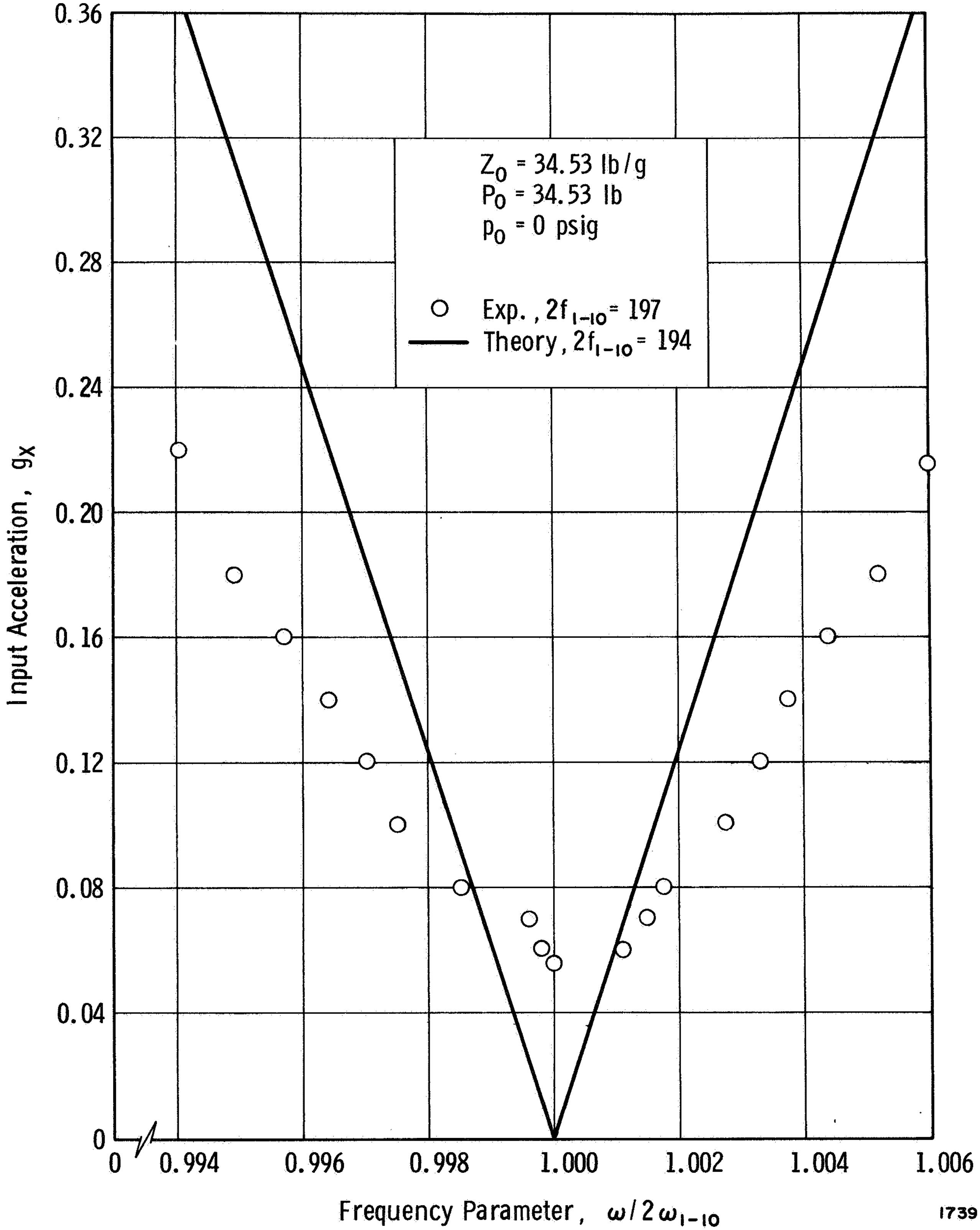
impedance Z_0 of the boundary condition at the tank top, could be measured. The use of acceleration impedance (force/acceleration) proved to be most convenient in this application. Variation of the impedance was achieved by using different rigid masses as well as the loading frame. The cylinder is made of 0.005-in. stainless steel, has a diameter of 10 in., and is 14.5 in. long (the same cylinder as that used in Reference 4).

Theoretical and experimental stability boundaries are compared in Figure 3 for the $k = 1$, $m = 10$ mode. Theoretical results are obtained from Equation (29) with $N = 5$ terms. Excitation conditions at or above the boundaries result in a principal parametric resonance whose mode shape is dominantly the $k = 1$, $m = 10$ natural mode, and whose frequency of motion is 1/2-subharmonic to the excitation. Experimental points were determined as the points of least acceleration where the parametric response would occur. It is apparent that significant deviation exists between theoretical and experimental results for the empty tank, and better agreement is achieved for greater liquid depths. After careful scrutiny, it was ascertained that the wider experimental stability boundaries are principally caused by imperfections in the cylinder. That is, split natural modes[9] and spatially shifting modal patterns occurred so that one exact natural frequency did not exist. As a result, the experimental system shows a tendency to be more unstable than predicted by theory. This trend is apparent in all the data. It is possible that somewhat better agreement could be achieved by the use of some form of imperfection theory in the analysis. This possibility remains to be investigated.



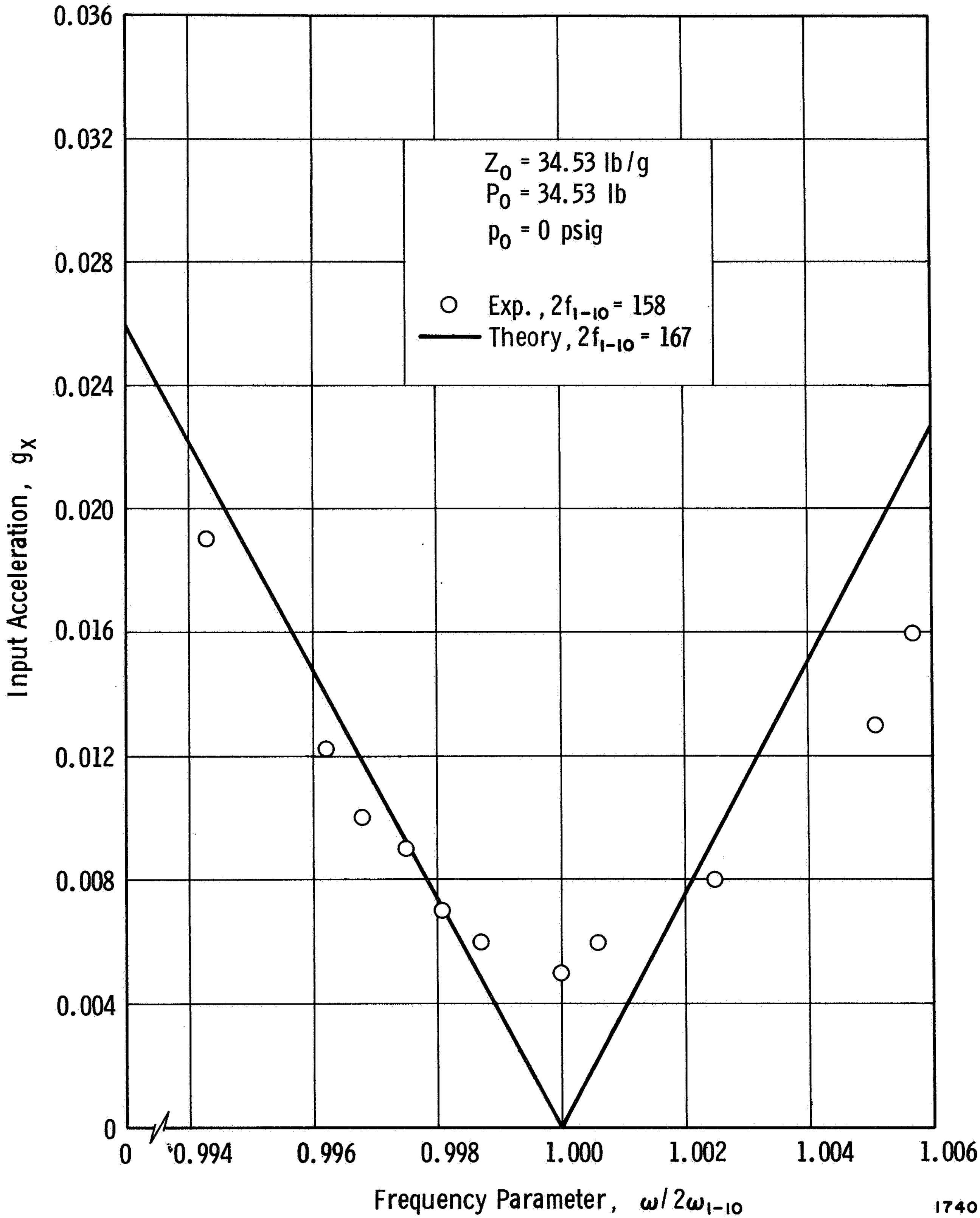
a. $h/\lambda = 0$

Figure 3. Influence Of Liquid Depth On Stability



b. $h/\lambda = 0.414$

Figure 3. Influence Of Liquid Depth On Stability



c. h/λ = 0.983

Figure 3. Influence Of Liquid Depth On Stability

It was desirable to determine the influence of the various system parameters on the stability boundaries for a given mode. This was done in terms of dimensional variables, in order to emphasize the complexity of this influence. For this purpose, it is necessary to understand the effects of the same parameters on the natural frequencies of the system. For convenience, some natural frequencies which were determined in the earlier work[4] for several symmetric and one nonsymmetric mode are given as functions of liquid depth in Figure 4.

It is recognized that, in general, a more unstable system will possess a stability boundary whose acceleration ordinate is at a lower value for a given value of the frequency parameter $2\omega_{1-10}$. Therefore, in order to assess the effects of axial load, ullage pressure, liquid depth, and top impedance, a stability boundary acceleration g_{x1} was determined at an excitation frequency value of $\omega_{x1} = 0.996\ (2\omega_{1-10})$ for a range of each of these parameters. Theoretical and experimental results are compared in Figures 5 through 8. These results must be compared with those in Figure 4 for proper interpretation. At a given liquid depth, increasing axial tension has only a small effect on natural frequencies and, likewise, only an insignificant effect on stability as shown in Figure 5. On the other hand, increasing ullage pressure significantly raises the natural frequencies of the nonsymmetric modes but leaves those of the lower symmetric modes essentially unchanged. Thus, as ω_{x1} approaches a natural frequency for a symmetric mode, the parametric excitation of the initial state is amplified, and the system becomes more unstable. This is reflected by the dips in the curves in

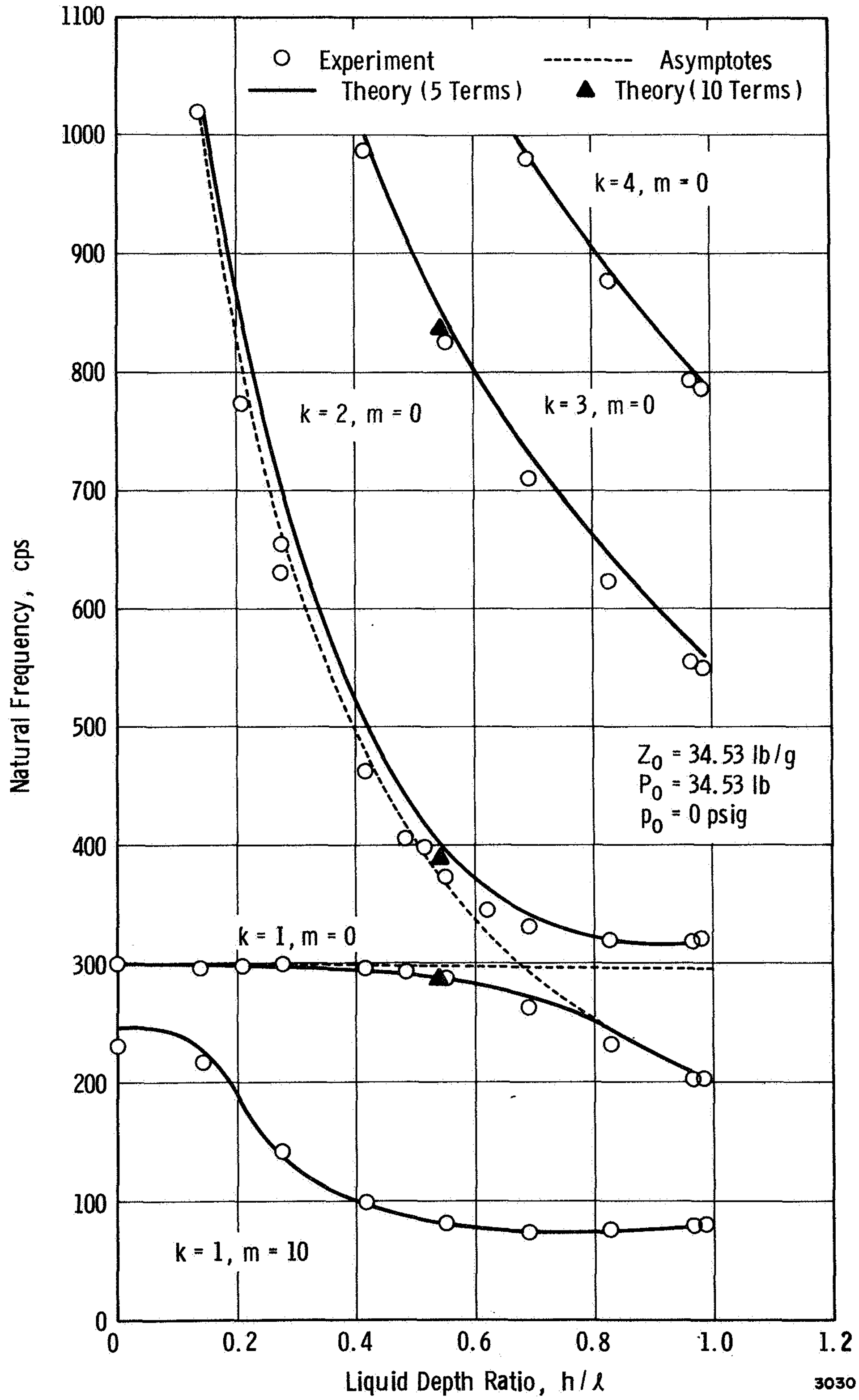


Figure 4. Natural Frequencies Of Partially Filled Tank

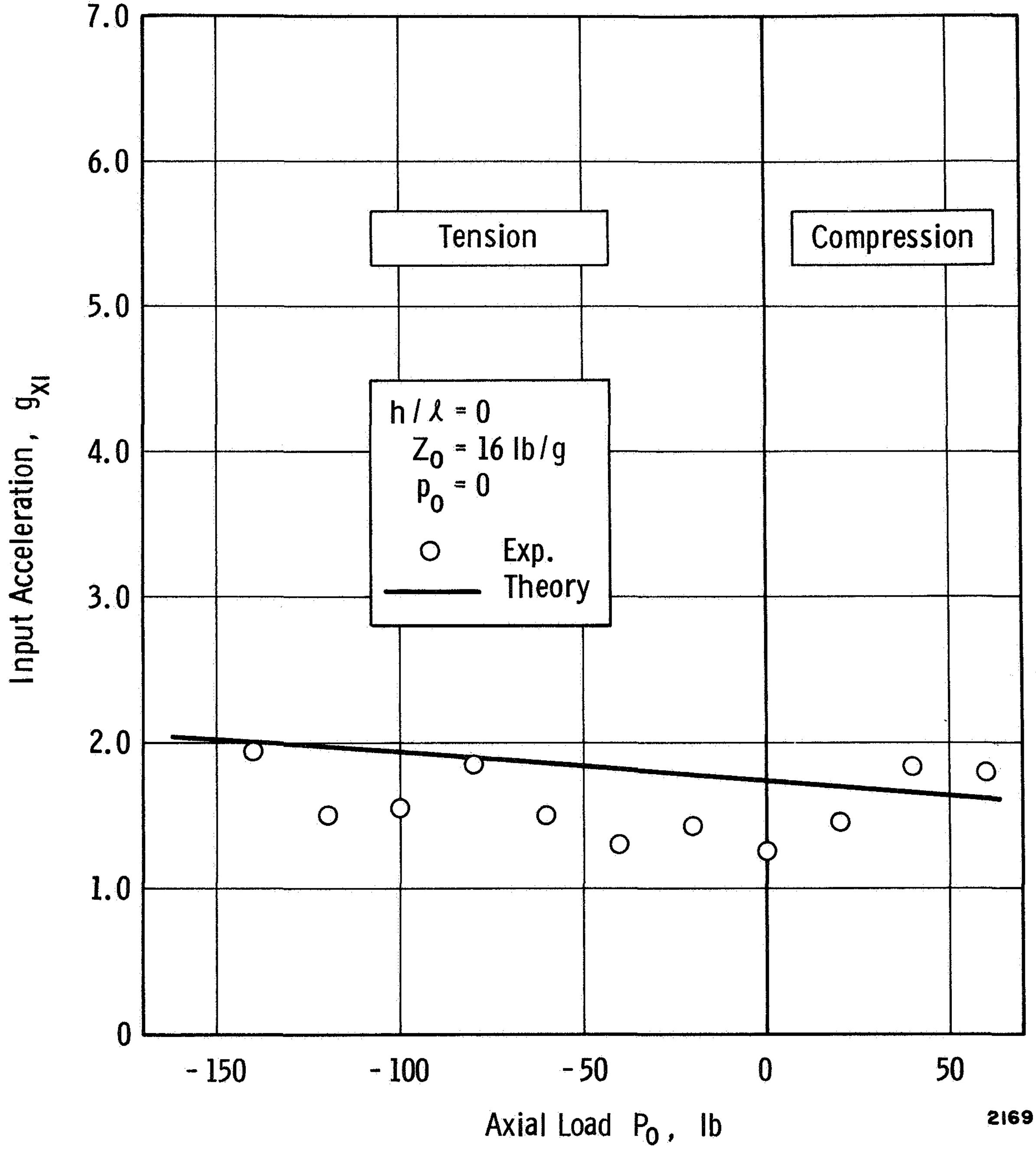


Figure 5. Influence Of Axial Load On Stability

Figure 6. It is also interesting to note that, at certain frequencies, the system becomes completely stable where the parametric coefficient in Equation (27) becomes zero.

Increasing liquid depth changes all natural frequencies, as shown in Figure 4, and has a profound influence on stability throughout the depth range, as shown in Figure 7. This results from the coincidence of ω_{x1} with natural frequencies of symmetric modes at certain points, as well as the provision of an increased distributed parametric loading on the tank wall.

The influence of top impedance on stability is shown in Figure 8. Increasing this impedance lowers the frequencies of symmetric modes while leaving the nonsymmetric mode frequencies unaltered. Thus, strong interaction can again be seen to occur. The dip in the curve occurs at an impedance such that ω_{x1} coincides with the natural frequency of the first symmetric mode.

It is obvious that variation of the above parameters can cause either an increase or decrease of stability, depending on the range of analysis. Further, it must be recognized that many nonsymmetric modes are present in the frequency range indicated in Figure 4, and each mode can become unstable as the one which was studied. Therefore, a complex pattern of instability and parametric resonance occurs with many overlapping regions of instability. The overall trend of the data shows good qualitative agreement between theory and experiment, although significant quantitative discrepancies exist because of the reasons previously discussed.

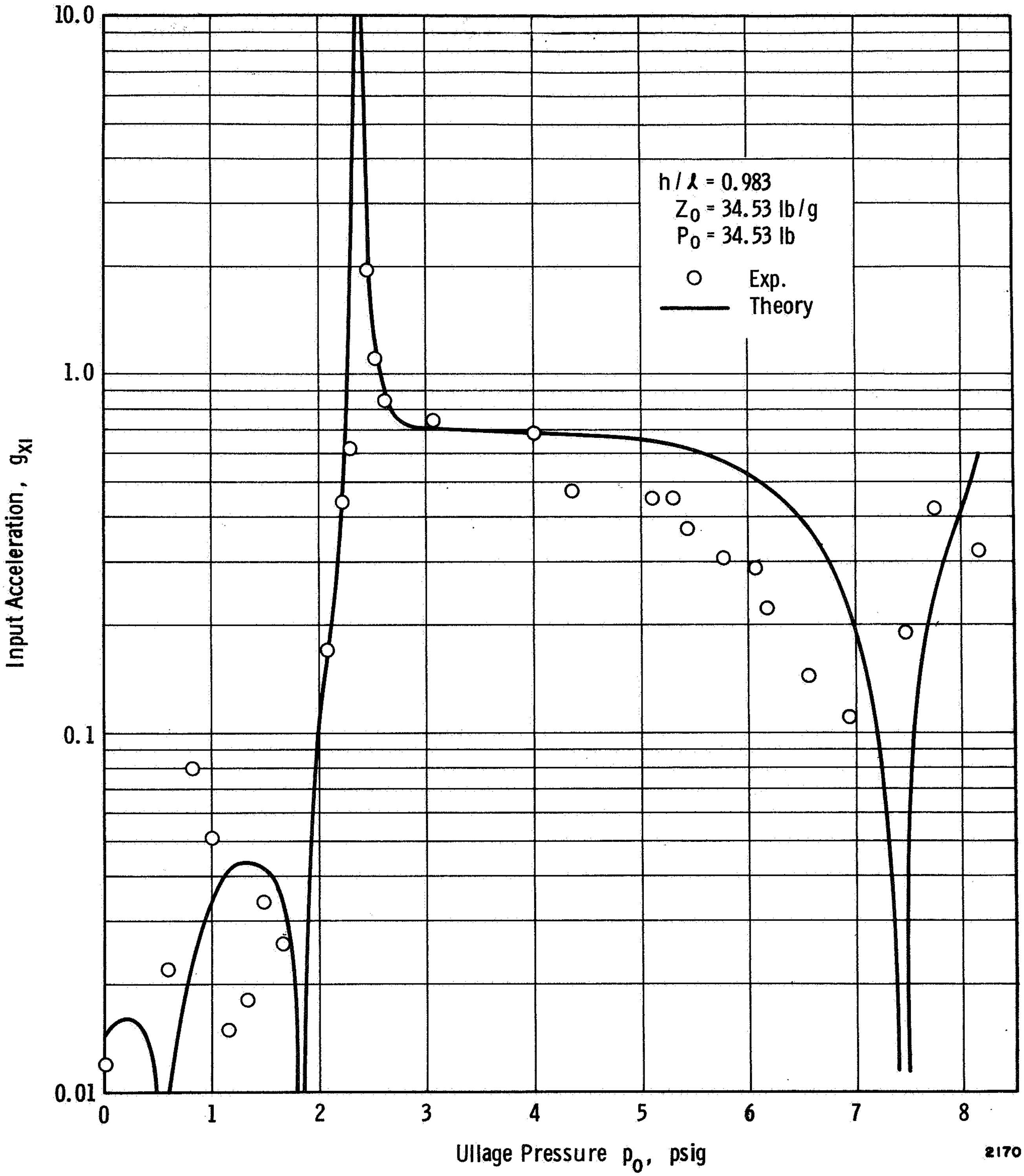


Figure 6. Influence Of Ullage Pressure On Stability

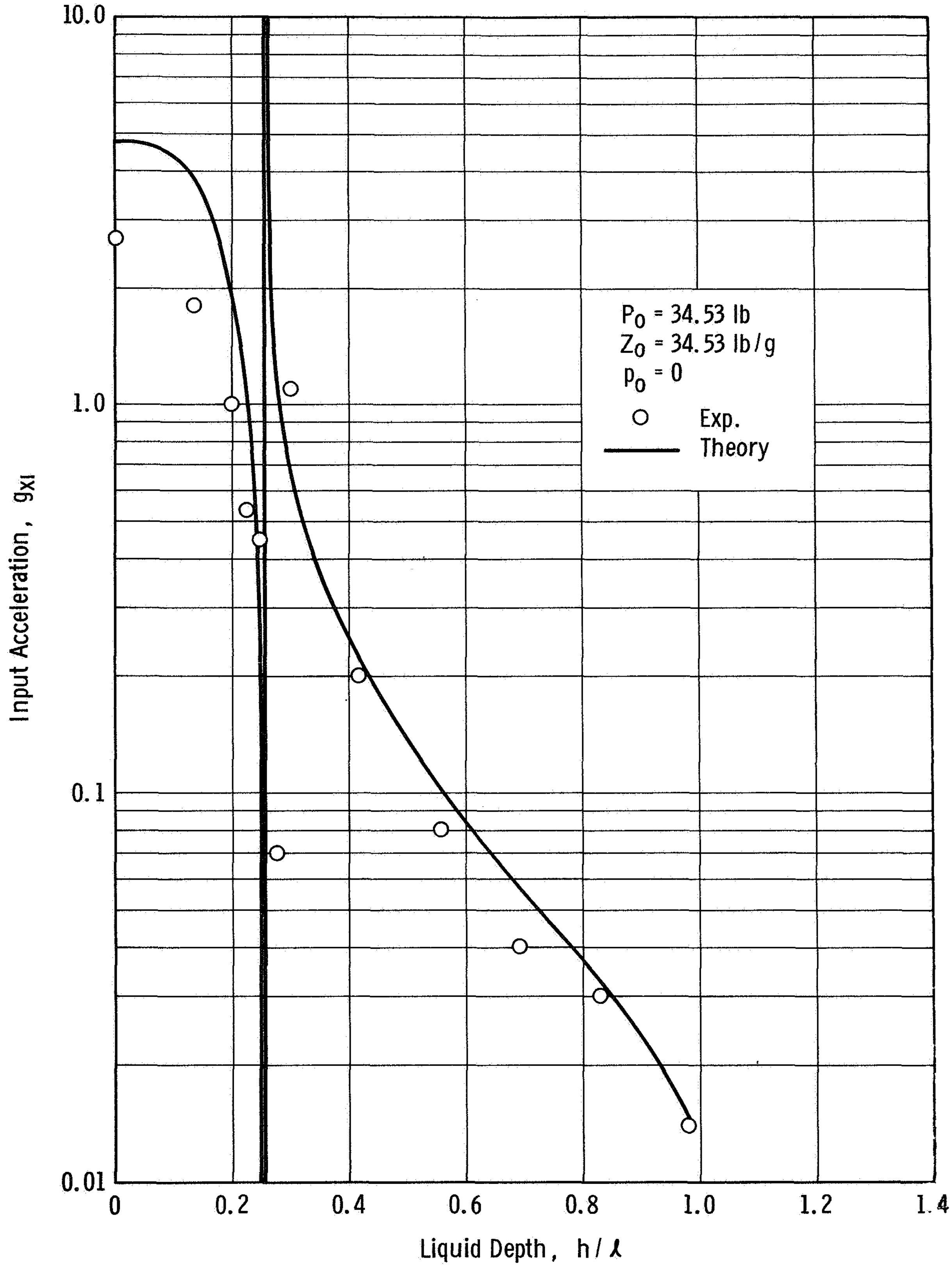


Figure 7. Influence Of Liquid Depth On Stability

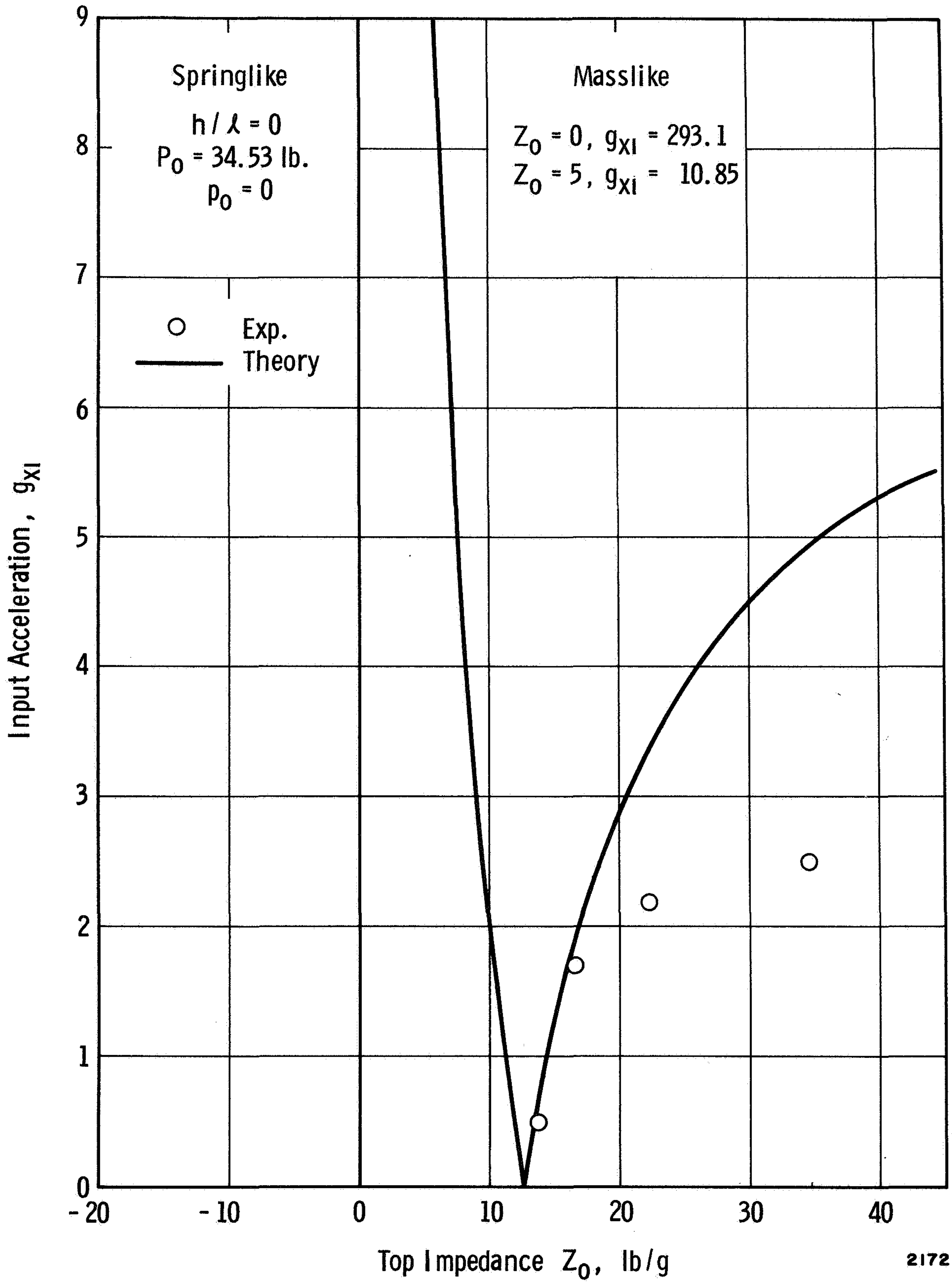


Figure 8. Influence Of Top Impedance On Stability

ACKNOWLEDGMENTS

The authors wish to express their sincere appreciation to several colleagues for assistance during the conduct of this research program. Particular mention should be given to Dr. H. Norman Abramson for his counsel, to Mr. Dennis Scheidt for performing most of the experiments, to Mr. Robert Gonzales for digital computer programming, and to Mr. Victoriano Hernandez for preparing the figures in this report.

REFERENCES

1. Bolotin, V. V., The Dynamic Stability of Elastic Systems, Holden-Day, Inc., San Francisco, California (1964).

2. Kana, D. D. and Gormley, J. F., "Longitudinal Vibration of a Model Space Vehicle Propellant Tank," AIAA Journal of Spacecraft and Rockets, Vol. 4, No. 12, December 1967, pp. 1585-1591.

3. Kana, D. D. and Craig, R. R., Jr., "Parametric Oscillations of a Longitudinally Excited Cylindrical Shell Containing Liquid," AIAA Journal of Spacecraft and Rockets, Vol. 5, No. 1, January 1968, pp. 13-21.

4. Kana, D. D. and Chu, W. H., "Influence of a Rigid Top Mass on the Response of a Pressurized Cylinder Containing Liquid," Final Report Part I, Contract No. NAS8-21282, SwRI Project No. 02-2332, June 2, 1969, also AIAA Jour. Spacecraft and Rockets, 6, 2, February 1969, pp. 103-110.

5. Sanders, J. L., Jr., "Nonlinear Theories for Thin Shells," Q. Appl. Math., Vol. XXI, No. 1 (1963), pp. 21-36.

6. Sanders, J. L., Jr., "An Improved First Approximation Theory for Thin Shells," NASA Rept. 24, June 1959.

7. Duncan, W. J., "Galerkin's Method in Mechanics and Differential Equations," Aeronautical Research Committee Reports and Memoranda No. 1798, London (1937).

8. Cunningham, W. J., Introduction to Nonlinear Analysis, McGraw-Hill Book Co., Inc., New York, N. Y. (1958).

9. Tobias, S. A., "A Theory of Imperfection for the Vibration of Elastic Bodies of Revolution," Engineering, 172, pp. 409-411 (1951).

DIGITAL COMPUTER PROGRAM

INPUT DATA DESCRIPTION

Card No.	Fortran Symbol	Variable† Name	Units	Definition
1	RHO	386ρ	$lb/in.^3$	weight density of liquid
	RHOS	$386\rho_s$	$lb/in.^3$	weight density of the shell
	A0	a_0	in.	inner radius of the tank
	SH	h	in.	depth of liquid
	SHS	h_s	in.	thickness of shell
	SL	ℓ	in.	length of the shell
2	P0	p_0	$lb/in.^2$	ullage pressure
	ENU	ν		Poissons ratio
	E	E	$lb/in.^2$	modulus of elasticity
	BMSTR	Z^{**}		nondimensional top impedance
	C0	c_0	$in./sec^2$	speed of sound in the liquid
3	NJ			no. of roots μ_{mj}
	N	N		no. terms in series expressions
4	W	$\omega/2\pi$	cps	excitation frequency
	BM	Z_0	$lb\,sec^2/in.$	top impedance
	CP0	P_0	lb	applied force
	NOPT			print option
	M	m		circumferential wave number
5	UMJ(I)	μ_{mj}		roots of $J'_m(\mu_{mj}) = 0,\ m \neq 0$
6	W	$\omega/2\pi$	cps	excitation frequency
	BM	Z_0	$lb\,sec^2/in.$	top impedance

†Note that some variables in computer program are slightly different from those as defined in NOMENCLATURE on pp. v to vi of text portion of this report.

Card No.	Fortran Symbol	Variable Name	Units	Definition
	CP0	p_0	lb	applied force
	NOPT			print option
	M	m		circumferential wavenumber
7	UMJ(I)	μ_{mj}	m = 0	roots of $J'_m(\mu_{mj}) = 0$, $m = 0$

PROGRAM OUTPUT

Printed Output

1. Input data h, h_s, ℓ, a_0, a, p_0, Z^{**}, 386ρ, $386\rho_s$, E, c_0, ν

2. In subroutine MITERS the mode no., eigenvalue, no. of iterations, no. of times Aitken's delta process is used, the eigenvector, and check eigenvalue and eigenvector

3. Mode no., $\tilde{\Omega}_k$, ω_k in rad/sec, and ω_k in cps

4. ω in cps, ω_k in cps, $\overline{K}$, $\overline{T}$, $\overline{M}$, $\overline{a}$, $\overline{q}$, $|X_0|$, g_x, and $\omega/2\omega_k$

PROGRAM NOTES

Subprograms Used

In addition to the main program the following subroutines are used:

1. BES, computes the Bessel functions J_n or I_n.

2. MATINV, computes the inverse of a real matrix.

3. MPRINT, prints matrix in matrix format.

4. MITERS, computes the eigenvalues and eigenvectors of a real or complex matrix by the power method.

5. NOPT is a print option that allows printing of intermediate results.

The following subroutines are included as a part of subprogram MITERS:

SWEEPX

NPNRMX

DPMLTX

SYMBOLIC LISTING

Some of the program FORTRAN symbols which were not defined in the Input Data Description are:

Fortran Symbol	Variable† Name	
A	a	
BMS	$M_s = 2\pi\rho_s a h_s \ell$	
HS	H_s	
CS	c_s	
H	H	
SLSTR	ℓ / a	
AONSQ	$\alpha^2_{n'}$	
X10	χ_{10}	
X20	χ_{20}	
ALN(I)	λ_n	$n = 1, N$
ALNP(I)	$\lambda_{n'}$	$n' = 1, N$
X1(I)	$\chi_{1n'}$	$n' = 1, N$
X2(I)	$\chi_{2n'}$	$n' = 1, N$
X0B(I)	$\tilde{\chi}_{0n'}$	$n' = 1, N$
X1B(I)	$\tilde{\chi}_{1n'}$	$n' = 1, N$
ENB(I)	$\tilde{E}_{n'}$	$n' = 1, N$

†Corresponding to Final Report Part I.

Fortran Symbol	Variable Name	
ETA(J)	η_{mj}	$j = 1, NJ$
CJH(I)	C_{jH}	$j = 1, NJ$
CNJ(I, J)	$\tilde{C}_{n'j}$	$n' = 1, N$; $j = 1, NJ$
BMJN(I, J)	$\tilde{B}_{mjn}$	$j = 1, NJ$; $n = 1, N$
BN1(I)	N_{1mn}	$n = 1, N$
BN2(I)	N_{2mn}	$n = 1, N$
BN0(I)	N_{0mn}	$n = 1, N$
BOON(I)	$\tilde{B}_{00n}$	$n = 1, N$
BMMNN(I, J)	$M_{mn'n}$	$n' = 1, N$; $n = 1, N$
R1M(I, J)	$R_{1mn'n}$	
S2M(I, J)	$S_{2mn'n}$	
U1M(I, J)	$U_{1mn'n}$	
U2M(I, J)	$U_{2mn'n}$	
T3M(I, J)	$T_{3mn'n}$	
U3M(I, J)	$U_{3mn'n}$	
V2M(I, J)	$V_{2mn'n}$	
W1M(I, J)	$W_{1mn'n}$	
W2M(I, J)	$W_{2mn'n}$	
W3M(I, J)	$W_{3mn'n}$	
R2M(I, J)	$R_{2mn'n}$	
R3M(I, J)	$R_{3mn'n}$	
S1M(I, J)	$S_{1mn'n}$	

Fortran Symbol	Variable Name
S3M(I, J)	$S_{3mn'n}$
T1M(I, J)	$T_{1mn'n}$
T2M(I, J)	$T_{2mn'n}$
V1M(I, J)	$V_{1mn'n}$
X1M(I, 1)	$X_{1n'}$
U4M(1, I)	U_{4n}
V5M(1, I)	V_{5n}
R4M(1, I)	R_{4n}
R5M(1, I)	R_{5n}
R6M(1, I)	R_{6n}
O1M(I, 1)	$O_{1n'}$
P1M(I, 1)	$P_{1n'}$
Y1M(I, 1)	$Y_{1n'}$
Z1M(I, 1)	$Z_{1n'}$
Y2M(I, 1)	$Y_{2n'}$
Z2M(I, 1)	$Z_{2n'}$
Y3M(I, 1)	$Y_{3n'}$
Z3M(I, 1)	$Z_{3n'}$
P2M(I, 1)	$P_{2n'}$
Q2M(I, 1)	$Q_{2n'}$
V4M(1, I)	V_{4n}
S5M(1, I)	S_{5n}

Fortran Symbol	Variable Name
V3M(I, J)	$V_{3mn'n}$
V5M(1, I)	V_{5n}
S5M(1, I)	S_{5n}
UVW(I, J)	$[U]$
RST(I, J)	$[R]$
UTR(I, J)	$[U]^{-1}[R]$
UWR(I, J)	$\left[[U] - \tilde{\Omega}^2[R]\right]^{-1}$
AI(I)	$I_{0n'}$
QBH(I)	$\hat{Q}^B_{0n'}$
FH(I)	$F_{rn'}$
APH(I)	$[\{A^p_{mn}\}\{B^p_{mn}\}\{C^p_{mn}\}B^p_{mo}B^p_1B^p_2]'$
B0PH	B^p_0
B1PH	B^p_1
B2PH	B^p_2
CN1(I)	$N_{1n''}$
CN2(I)	$N_{2n''}$
CN3(I)	$N_{3n''}$
DNNN(I, J, K)	$d_{n''n'n}$
ENNN(I, J, K)	$e_{n''n'n}$
E0NN(I, J)	$e_{0n'n}$
TPNN(I, J)	$T'_{n'n_k}$
BMBAR	$\overline{M}$

Fortran Symbol	Variable Name
BKBAR	$\overline{K}$
BTBAR	$\overline{T}$
ABAR	$\overline{a}$
QBAR	$\overline{q}$
X0AQ	$\lvert(1 - \overline{a})/\overline{q}\rvert$
X0STR	$\lvert(1 - \overline{a})/\overline{q}\rvert\omega^2 a/g$
WSTR	$\omega/2\omega_k$
WKBSQ	$\tilde{\Omega}^2$
WKSQ	$\tilde{\Omega}_k^2$
AMNK(I, J)	$A_{mn'k}$

COMPUTATIONS LABORATORY — SOUTHWEST RESEARCH INSTITUTE

PROBLEM: EXAMPLE KEYPUNCH FORM — PAGE 1 OF 2

PROGRAMMER:

DATE

FORTRAN STATEMENT

```
      ρ,ρs,a0,h,hs,ℓ      (FΦRMAT 7F10.0)
0.0361    0.283     5.        14.25     0.005     14.75
      p0,ν,E,M**,c0       (FΦRMAT 2F10.0,3E12.3)
1.125     0.3            2.900+07    1.000+30    5.600+04
      NJ,N                (FΦRMAT 2I5)
    5    5
      ω,M,P0,NΦPT,m       (FΦRMAT 3F10.0,2I5)
300.0     34.53     34.53         0   10
      μmj, m=10, j=1,5    (FΦRMAT 5F15.0)
11.77088       16.44785       20.22304       23.76071       27.18202
      ω,M,P0,NΦPT,m       (FΦRMAT 3F10.0,2I5)
300.0     34.53     34.53         0    0
```

COMPUTATIONS LABORATORY — SOUTHWEST RESEARCH INSTITUTE

PROBLEM:

PROGRAMMER:

PAGE 2 OF 2

DATE

FORTRAN STATEMENT

```
      μmi, m=0, j=1,5  (FØRMAT 5F15.0)
3.8317060      7.0155867      10.1734681     13.3236916     16.4706301
```

```
      PROGRAM SLOSH (INPUT,OUTPUT,TAPE60=INPUT)
C     DETERMINATION OF NATURAL FREQUENCIES AND MODE SHAPES
C     OF A PARTIALLY FILLED SHELL
C     PROJECT  0 2 - 1 8 7 6
C     SEPTEMBER 1967
C     MODIFIED 4 SEPTEMBER 1968
C     PROJECT  0 2 - 2 3 3 2
      COMMON PI,H/COM1/A,A0,CSCOW/COM2/UMJ( 5)/COM3/AONSQ/COM4/ALN(10)/
     1  COM5/ALNP(10)
      DIMENSION X1(10),X2(10),X0B(10),X1B(10),        ENB(10),ETA( 5),
     1  CJH( 5),CNJ(10, 5),BMJN( 5,10),BN0(10),BN1(10),BN2(10),
     2  BOON(10),BMMNN(10,10),R1M(10,10),R2M(10,10),R3M(10,10),
     3  S1M(10,10),S2M(10,10),S3M(10,10),T1M(10,10),T2M(10,10),
     4  T3M(10,10),U1M(10,10),U2M(10,10),U3M(10,10),V1M(10,10),
     5  V2M(10,10),V3M(10,10),W1M(10,10),W2M(10,10),W3M(10,10),
     6  UVW(33,33),RST(33,33),UWR(33,33),T(1000),UTRSV(33,36)
      DIMENSION U4M(1,10),V4M(1,10),W4M(1,10),U5M(1,10),V5M(1,10),
     1 W5M(1,10),U6M(1,10),V6M(1,10),W6M(1,10),X1M(10,1),Y1M(10,1),
     2 Z1M(10,1),X2M(10,1),Y2M(10,1),Z2M(10,1),X3M(10,1),Y3M(10,1),
     3 Z3M(10,1),R4M(1,10),S4M(1,10),T4M(1,10),R5M(1,10),S5M(1,10),
     4 T5M(1,10),R6M(1,10),S6M(1,10),T6M(1,10),O1M(10,1),P1M(10,1),
     5 Q1M(10,1),O2M(10,1),P2M(10,1),Q2M(10,1),O3M(10,1),P3M(10,1),
     6 Q3M(10,1),UTR(33,36),BMMNNK(4,10,10),BMMNN0(4,10,10)
      DIMENSION  IROW(34),ICOL(33),AI(10),QBH(10),FH(33),APH(33)
      DIMENSION  WCPS(4),AMNK(33,4),CN1(10),CN2(10),CN3(10),
     1 DNNN(10,10,10),ENNN(10,10,10),EONN(10,10),TPNN(10,10)
      DIMENSION           GUESS(33,1),VECTOR(33, 4),EIGVAL( 4),
     1  NITER( 4),US(33,8),HH(33,29),NAKSR( 4),EIGCPS( 5)
      PI = 3.14159265
      G = 386.0
      EPS = 1.0E-04
      SMLST = 1.0E-07
      NGUESS = 0
      NMODE = 1
      NITRSP = 100
      EPSP = 0.5E-08
      AITKEN = .9
      NTAPE = 1
 1000 READ 200, RHO,RHOS,A0,SH,SHS,SL
  200 FORMAT (7F10.0)
      IF (EOF,60)700,705
  700 STOP
  705 READ 205, P0,ENU,E,BMSTR,C0
  205 FORMAT (2F10.0,3E12.3)
      READ 215, NJ,N
  215 FORMAT(3I5)
      N3 = 3*N+3
      NDIM = 33
      RHO = RHO/386.
      RHOS = RHOS/386.
      A = A0+0.5*SHS
      BMS = 2.0*PI*RHOS*A*SHS*SL
      HS = SHS/A
      CS = SQRTF(E/RHOS)
      H = SH/A
      SLSTR = SL/A
      AONSQ = 0.5*SLSTR
      X10 = 0.5*SLSTR
      X20 = SLSTR**2/3.0
      PRINT 505
  505 FORMAT (*1 PROJECT  0 2 - 2 3 3 2 *)
      PRINT 530,    SH,SHS,SL,A0,A,P0,BMSTR,RHO,RHOS,E,C0,ENU
```

```
  530 FORMAT(                 *0   H = *,E10.3,* (IN)*,17X,*HS = *,E10.3,   SL006400
     1* (IN)*,16X,*L = *,E10.3,* (IN)*//3X,*A0 = *,E10.3,* (IN)*,18X,      SL006500
     2*A = *,E10.3,* (IN)*//*   P0 = *,E10.3,18H (LB-SEC**2/IN**4),3X,     SL006600
     36HM** = ,E10.3//*  RHO = *,E10.3,18H (LB*SEC**2/IN**4),2X,           SL006700
     4 *RHOS = *,E10.3,18H (LB-SEC**2/IN**4)//4X, *E = *,E10.3,            SL006800
     5 11H (LB/IN**2),11X,*C0 = *,E10.3,* (IN/SEC)*,11X,*NU = *,E10.3)    SL006900
      KOPT = 0                                                             SL007000
 2000 READ 210, W,BM,CP0,NOPT,M                                            SL007100
  210 FORMAT (3F10.0,2I5)                                                  SL007200
      READ 220, (UMJ(I),I=1,NJ)                                            SL007300
  220 FORMAT (5F15.0)                                                      SL007400
      BM = BM/386.                                                         SL007500
      IF (M)3,4,3                                                          SL007600
    4 BMSTR = (BM*(1.0-ENU**2))/(2.0*PI*RHOS*A**3*HS)                      SL007700
      JTER = 1                                                             SL007900
      W = .996*(2.*WCPS(JTER))                                             SL008000
      WRAD = 2.*PI*W                                                       SL008100
      WKB = WCPS(JTER)                                                     SL008200
      WKSQ = (1.-ENU**2)*(2.*PI*WKB)**2*(A/CS)**2                          SL008300
      GO TO 2                                                              SL008400
    3 CONTINUE                                                             SL008500
      DO 7000 ITER=1,1                                                     SL008600
      IF (KOPT)6000,6005,6000                                              SL008700
 6000 W = .996*(   WCPS(ITER))                                             SL008800
      WKB = WCPS(ITER)                                                     SL008900
      WKSQ = (1.-ENU**2)*(2.*PI*WKB)**2*(A/CS)**2                          SL009000
 6005 WRAD = 2.*PI*W                                                       SL009100
    2 OMEGA = (WRAD*A)/CS                                                  SL009200
      WSQ = OMEGA**2                                                       SL009300
      WKBSQ = (1.0-ENU*ENU)*WSQ                                            SL009400
      CSCOW = (CS/C0)*OMEGA                                                SL009700
      DO 10 I=1,N                                                          SL009800
      ALN(I) = (I*PI*A)/SL                                                 SL009900
      ALNP(I) = (I*PI*A)/SL                                                SL010000
      X1(I) = (2.0/(SLSTR*ALNP(I)**2))*(-1.0+(-1)**I)                      SL010100
      X2(I) = (4.0/ALN(I)**2)*(-1)**I                                      SL010200
      X0B(I)  = (2.0/(SLSTR*ALNP(I)))*(1.0-(-1)**I)                        SL010300
      X1B(I) = (2.0*(-1)**(I-1))/ALNP(I)                                   SL010400
   10 CONTINUE                                                             SL010500
      DO 5 I=1,N                                                           SL010600
      TM1 = ALNP(I)+CSCOW                                                  SL010700
      TM2 = ALNP(I)-CSCOW                                                  SL010800
      ENB(I) = (((1.0-COSF(TM1*H))/TM1)+((1.0-COSF(TM2*H))/TM2))/(2.0*     SL010900
     1  A0NSQ)                                                             SL011000
    5 CONTINUE                                                             SL011100
      DO 15 I=1,NJ                                                         SL011200
      ETA(I) = SQRTF(ABSF(UMJ(I)**2-CSCOW**2))                             SL011300
      IF (UMJ(I)-CSCOW)20,25,25                                            SL011400
   25 TM1 = EXPF(ETA(I)*H)                                                 SL011500
      CJH(I) = 0.5*(TM1+(1.0/TM1))                                         SL011600
      GO TO 15                                                             SL011700
   20 CJH(I) = COSF(ETA(I)*H)                                              SL011800
   15 CONTINUE                                                             SL011900
      DO 30 I=1,N                                                          SL012000
      TM1 = ALNP(I)*H                                                      SL012100
      SN = SINF(TM1)                                                       SL012200
      CN = COSF(TM1)                                                       SL012300
      DO 30 J=1,NJ                                                         SL012400
      IF (UMJ(J)-CSCOW)35,40,40                                            SL012500
   40 TM2 = ETA(J)*H                                                       SL012600
      TM3 = EXPF(TM2)                                                      SL012700
      CSH = 0.5*(TM3+(1.0/TM3))                                            SL012800
```

```
      SNH = 0.5*(TM3-(1.0/TM3))
      CNJ(I,J) = (H/(AONSQ*(TM2**2+TM1**2)))*(TM1*CN*CSH-TM1-TM2*SN*SNH)
      GO TO 30
   35 TM2 = ALNP(I)-ETA(J)
      TM3 = ALNP(I)+ETA(J)
      CNJ(I,J) = (0.5/AONSQ)*(((1.0-COSF(TM3*H))/TM3)+((1.0-COSF(TM2*
     1  H))/TM2))
   30 CONTINUE
      DO 45 I=1,NJ
      DO 45 J=1,N
      SUMX = 0.
      SUMY = 1.0E-50
      K = 0
   60 TRM = ((-1)**K*DKN(K,J)*EMKJ(M,K,I))/CJH(I)
      SUMX = SUMX+TRM
      IF (ABSF((SUMX-SUMY)/SUMY)-EPS)50,50,55
   55 SUMY = SUMX
      K = K+1
      IF (K-100)60,50,50
   50 BMJN(I,J) = SUMY
   45 CONTINUE
      DO 65 I=1,N
      TM1 = (1.0-ENU*ENU)*WSQ
      TM2 = (A/SL)-0.5*BMSTR*WSQ
      TM3 =-(A/SL)*(-1)**I*BMSTR*WSQ
      TM4 =-(A/SL)*BMSTR*WSQ
      TM5 = 1.0-(A/SL)*X10
      TM6 = 1.0-(M**2/(2.0*(1.0+ENU)*WSQ))
      TM7 = 0.5*(1.0-ENU)*M*M*X20
      BN0(I) = -((A/SL)**2*((2.0*TM1*X20-TM7)/(2.0*TM1))*(TM3/TM2)-TM5*
     1  TM6)/(1.0+((2.0*TM1*X20-TM7)/(2.0*TM1))*(TM4/TM2)*(A/SL)**2-
     2  TM5*TM6)
      BN1(I)   = -(A/SL)*(1.0+BN0(I))
      BN2(I) = ((TM4*BN0(I)+TM3)*(A/SL)**2)/TM2
   65 CONTINUE
      DO 66 I=1,N
      SUMX = 0.
      SUMY = 1.0E-50
      K = 1
   69 TRM = (-1)**K*DKN(K,I)*EMKJ(M,K,0)
      SUMX = SUMX+TRM
      IF (ABSF((SUMX-SUMY)/SUMY)-EPS)67,67,68
   68 SUMY = SUMX
      K = K+1
      IF (K-100)69,67,67
   67 BOON(I) = SUMY/COSF(CSCOW*H)
   66 CONTINUE
      DO 70 I=1,N
      DO 70 J=1,N
      SUMX = 0.
      SUMY = 1.0E-50
      IF (M)86,86,87
   86 K = 1
      GO TO 85
   87 K=0
   85 TRM = RMK(M,K)*DKN(K,J)*DKNB(K,I)
      SUMX = SUMX+TRM
      IF (ABSF((SUMX-SUMY)/SUMY)-EPS)75,75,80
   80 SUMY = SUMX
      K = K+1
      IF (K-100)85,85,75
   75 SUMZ = 0.
```

```
      DO 90 L=1,NJ
      CALL BFS(M,UMJ(L),0,XJ0,T)
      TRM = XJ0*BMJN(L,J)*CNJ(I,L)
      SUMZ = SUMZ+TRM
   90 CONTINUE
      IF (M)91,92,91
   92 CALL BFS(0,CSCOW,0,XJ0,T)
      CALL BFS(1,CSCOW,0,XJ1,T)
      TRM = -BOON(J)*ENB(I)+(((2.0*ENB(I))/(CSCOW**2*COSF(CSCOW*H)))-
     1  ((XJ0*DKNB(0,I))/(CSCOW*XJ1)))*DKN(0,J)
         BMMNN(I,J) = ((RHO*A)/(RHOS*SHS))*(SUMY-SUMZ+TRM)
      GO TO 70
   91 TRM = 0.
         BMMNN(I,J) = ((RHO*A)/(RHOS*SHS))*(SUMY-SUMZ+TRM)
      IF ( ITER )93,70,93
   93 CONTINUE
      IF (KOPT)6010,6015,6010
 6010 BMMNN0(ITER,I,J) = BMMNN(I,J)
      GO TO 70
 6015 BMMNNK(ITER,I,J) = BMMNN(I,J)
   70 CONTINUE
      IF ( M )6025,6020,6025
 6025 CONTINUE
      IF (KOPT)2000,6020,2000
 6020 CONTINUE
      DO 95 I=1,N
      DO 95 J=1,N
      IF (I-J)100,105,100
  105 R1M(I,J) = 1.0+BMMNN(I,J)
      GO TO 95
  100 R1M(I,J) = BMMNN(I,J)
   95 CONTINUE
      DO 110 I=1,N
      DO 110 J=1,N
      IF (I-J)115,120,115
  120 S2M(I,J) = 1.
      GO TO 110
  115 S2M(I,J) = 0.
  110 CONTINUE
      DO 125 I=1,N
      DO 125 J=1,N
      TM1 = 1.0+(HS*HS*(ALN(J)**2+M*M )**2)/12.
      TM2 = (P0/E)*(0.5*ALN(J)**2+M*M)
      IF (I-J)130,135,130
  135 X3N = (A/SL)*(.5*H**2-((1.-COS(2.*ALN(J)*H))/(4.*ALN(J)**2)))
      TM3 = X3N*M*M*(RHO*G*A/E)
      U1M(I,J) = TM1+(TM2+TM3-((CP0 *ALN(J)**2)/(2.0*PI*A*A*E)))*((1.0-
     1  ENU**2)/HS)
      GO TO 125
  130 TM3 = ALN(J)-ALNP(I)
      TM4 = ALN(J)+ALNP(I)
      X3N = (A/SL)*(((1.-COS(TM3*H))/TM3**2)-((1.-COS(TM4*H))/TM4**2))
      U1M(I,J) = (X3N*M*M)*((1.0-ENU**2)/HS)*(RHO*G*A/E)
  125 CONTINUE
      DO 140 I=1,N
      DO 140 J=1,N
      IF (I-J)145,150,145
  150 U2M(I,J) = -ENU*ALN(J)
      T3M(I,J) = 1.0
      U3M(I,J) = M
      V2M(I,J) = ALN(J)**2+0.5*(1.0-ENU)*M*M
      W1M(I,J) = M
```

```
      W2M(I,J) = -0.5*(1.0+ENU)*M*ALN(J)
      W3M(I,J) = 0.5*(1.0-ENU)*ALN(J)**2+M*M
      GO TO 140
  145 U2M(I,J) = 0.
      T3M(I,J) = 0.
      U3M(I,J) = 0.
      V2M(I,J) = 0.
      W1M(I,J) = 0.
      W2M(I,J) = 0.
      W3M(I,J) = 0.
  140 CONTINUE
      DO 155 I=1,N
      DO 155 J=1,N
      R2M(I,J) = 0.
      R3M(I,J) = 0.
      S1M(I,J) = 0.
      S3M(I,J) = 0.
      T1M(I,J) = 0.
      T2M(I,J) = 0.
  155 CONTINUE
      DO 160 I=1,N
      DO 160 J=1,N
      IF (I-J)161,162,161
  162 V1M(I,J) = -ALN(J)*ENU
      GO TO 160
  161 V1M(I,J) = 0.
  160 CONTINUE
      DO 165 I=1,N
      X1M(I,1) = X2M(I,1) = X3M(I,1) = 0.
      U4M(1,I) = U5M(1,I) = U6M(1,I) = 0.
      V5M(1,I) = W5M(1,I) = V6M(1,I) = W6M(1,I) = W4M(1,I) = 0.
      R4M(1,I) = S4M(1,I) = T4M(1,I) = 0.
      R5M(1,I) =            T5M(1,I) = 0.
      R6M(1,I) = S6M(1,I) = T6M(1,I) = 0.
      O1M(I,1) = O2M(I,1) = O3M(I,1) = 0.
      P1M(I,1) = P3M(I,1) = Q1M(I,1) = Q3M(I,1) = 0.
      Y1M(I,1) = ENU*X0B(I)
      Z1M(I,1) = ENU*X1B(I)
      Y2M(I,1) = M**2*(.5*(1.-ENU))*X1(I)
      Z2M(I,1) = M**2*(.25*(1.-ENU))*X2(I)
      Y3M(I,1) = M*(.5*(1.+ENU))*X0B(I)
      Z3M(I,1) = M*(.5*(1.+ENU))*X1B(I)
      P2M(I,1) = X1(I)
      Q2M(I,1) = .5*X2(I)
      V4M(1,I) = -1.
      S5M(1,I) = (-1)**I/(1.-ENU**2)
      DO 165 J=1,N
      IF (I-J)170,175,170
  175 V3M(I,J) =      -ALN(J) *0.5*M*(1.0+ENU)
      GO TO 165
  170 V3M(I,J) = 0.
  165 CONTINUE
      X5M =       Z4M = O4M = P4M = Q4M = P5M = 0.
      X4M = -1.
      Y4M = SL/A
      Y5M = 1./BMSTR
      Z5M = Y4M*Y5M
      O5M = 1./(1.-ENU**2)
      Q5M = (.5*(SL/A)**2)/(1.-ENU**2)
      Y6M = M**2*(.5*(1.-ENU    ))*(-(SL/A)+X10)
      Z6M = -(1.-.5*(M**2*(1.-ENU))*.5*X20)
      X6M = .5*M**2*(1.-ENU)
```

```
      O6M = 1.
      P6M = X10-SL/A
      Q6M = .5*X20
      IF (M)166,166,167
  167 X5M = 1.
      Y5M = 0.
      Z5M = .5*(SL/A)**2
      O5M = 0.
      P5M = 0.
      Q5M = 0.
      DO 168 I=1,N
      V5M(1,I) = (-1)**I
      S5M(1,I) = 0.
  168 CONTINUE
  166 DO 185 I=1,N
      JK = I+N
      JL = I+2*N
      IJ = 3*N+1
      IK = 3*N+2
      IL = 3*N+3
      UVW(I,IJ) = X1M(I,1)
      UVW(I,IK) = Y1M(I,1)
      UVW(I,IL) = Z1M(I,1)
      UVW(JK,IJ) = X2M(I,1)
      UVW(JK,IK) = Y2M(I,1)
      UVW(JK,IL) = Z2M(I,1)
      UVW(JL,IJ) = X3M(I,1)
      UVW(JL,IK) = Y3M(I,1)
      UVW(JL,IL) = Z3M(I,1)
      UVW(IJ,I) = U4M(1,I)
      UVW(IJ,JK)= V4M(1,I)
      UVW(IJ,JL)= W4M(1,I)
      UVW(IK,I)  = U5M(1,I)
      UVW(IK,JK) = V5M(1,I)
      UVW(IK,JL) = W5M(1,I)
      UVW(IL,I)  = U6M(1,I)
      UVW(IL,JK) = V6M(1,I)
      UVW(IL,JL) = W6M(1,I)
      DO 185 J=1,N
      IK = J+N
      IL = J+2*N
       UVW(I,J) = U1M(I,J)
       UVW(I,IK) = V1M(I,J)
       UVW(I,IL) = W1M(I,J)
       UVW(JK,J) = U2M(I,J)
       UVW(JK,IK) = V2M(I,J)
       UVW(JK,IL) = W2M(I,J)
       UVW(JL,J) = U3M(I,J)
       UVW(JL,IK) = V3M(I,J)
       UVW(JL,IL) = W3M(I,J)
  185 CONTINUE
      DO 190 I=1,N
      JK = I+N
      JL = I+2*N
      IJ = 3*N+1
      IK = 3*N+2
      IL = 3*N+3
      RST(I,IJ) = O1M(I,1)
      RST(I,IK) = P1M(I,1)
      RST(I,IL) = Q1M(I,1)
      RST(JK,IJ) = O2M(I,1)
      RST(JK,IK) = P2M(I,1)
```

```
      RST(JK,IL) = Q2M(I,1)
      RST(JL,IJ) = O3M(I,1)
      RST(JL,IK) = P3M(I,1)
      RST(JL,IL) = Q3M(I,1)
      RST(IJ,I) = R4M(1,I)
      RST(IJ,JK)= S4M(1,I)
      RST(IJ,JL)= T4M(1,I)
      RST(IK,I)  = R5M(1,I)
      RST(IK,JK) = S5M(1,I)
      RST(IK,JL) = T5M(1,I)
      RST(IL,I)  = R6M(1,I)
      RST(IL,JK) = S6M(1,I)
      RST(IL,JL) = T6M(1,I)
      DO 190 J=1,N
      IK = J+N
      IL = J+2*N
      RST(I,J) = R1M(I,J)
      RST(I,IK) =S1M(I,J)
      RST(I,IL) = T1M(I,J)
      RST(JK,J) = R2M(I,J)
      RST(JK,IK) = S2M(I,J)
      RST(JK,IL) = T2M(I,J)
      RST(JL,J) = R3M(I,J)
      RST(JL,IK) = S3M(I,J)
      RST(JL,IL) = T3M(I,J)
  190 CONTINUE
      IJ = 3*N+1
      IK = 3*N+2
      IL = 3*N+3
      UVW(IJ,IJ) = X4M
      UVW(IJ,IK) = Y4M
      UVW(IJ,IL) = Z4M
      UVW(IK,IJ) = X5M
      UVW(IK,IK) = Y5M
      UVW(IK,IL) = Z5M
      UVW(IL,IJ) = X6M
      UVW(IL,IK) = Y6M
      UVW(IL,IL) = Z6M
      RST(IJ,IJ) = O4M
      RST(IJ,IK) = P4M
      RST(IJ,IL) = Q4M
      RST(IK,IJ) = O5M
      RST(IK,IK) = P5M
      RST(IK,IL) = Q5M
      RST(IL,IJ) = O6M
      RST(IL,IK) = P6M
      RST(IL,IL) = Q6M
      IF (NOPT)4000,4005,4000
 4000 PRINT 4010
 4010 FORMAT (1H1,* ( U )  MATRIX *)
      CALL MPRINT (UVW,N3,N3,NDIM)
      PRINT 4015
 4015 FORMAT (1H1,* ( R )  MATRIX *)
      CALL MPRINT (RST,N3,N3,NDIM)
 4005 CONTINUE
      IF ( M )198,199,198
  198 CONTINUE
      CALL MATINV (UVW,IROW,ICOL,N3,NDIM,SMLST)
      DO 192 I=1,N3
      DO 192 J=1,N3
      SUM = 0.
      DO 191 K=1,N3
```

```
      SUM       = SUM       + UVW(I,K)*RST(K,J)
  191 CONTINUE
      UTR(I,J) = SUM
      UTRSV(I,J) = UTR(I,J)
  192 CONTINUE
      IF (NOPT)4020,4025,4020
 4020 PRINT 4030
 4030 FORMAT (1H1,* ( U )  MATRIX INVERSE *)
      CALL MPRINT (UVW,N3,N3,NDIM)
      PRINT 4035
 4035 FORMAT (1H1,* (U) INVERSE X (R) *)
      CALL MPRINT (UTR,N3,N3,NDIM)
 4025 CONTINUE
      CALL MITERS (UTR,NTAPE,N3,GUESS,NGUESS,NMODE,VECTOR,EIGVAL,
     1 NITER,NITRSP,EPSP,US,HH,MAXR,NC,AITKEN,NAKSR,UTRSV)
      DO 194 I=1,1
      XLDA = EIGVAL(I)
      TM1 = (CS/A)/SQRT(1.-ENU**2)
      OMEGR = SQRT(ABS(1./XLDA))*TM1
      OMEGC = OMEGR/(2.*PI)
      IF ( ITER )618,619,619
  619 EIGCPS(I) = OMEGC
  618 PRINT 615, I,XLDA,OMEGR,OMEGC
  615 FORMAT (I5,3E20.8)
  194 CONTINUE
      W = EIGCPS(ITER  )
      IF ( ITER )197,7000,197
  197 WCPS(ITER) = (SQRT(ABS(1./EIGVAL(ITER)))*TM1)/(2.*PI)
      DO 196 I=1,N3
         AMNK(I,ITER) = VECTOR(I,ITER)
  196 CONTINUE
 7000 CONTINUE
      KOPT = 1
      GO TO 3
  199 CONTINUE
      DO 195 I=1,N3
      DO 195 J=1,N3
      UWR(I,J) = UVW(I,J)-WKBSQ*RST(I,J)
  195 CONTINUE
      DO 600 I=1,N
      J = N+I
      K = 2*N+I
      AI(I) = ((-CSCOW*SINF(ALNP(I)*H)+ALNP(I)*SINF(CSCOW*H))/
     1  (ALNP(I)**2-CSCOW**2))/AONSQ
      QBH(I) = (RHO*A*OMEGA*C0*AI(I))/(RHOS*SHS*CS*COSF(CSCOW*H))
      FH(I) = (1.-ENU**2)*QBH(I)
      FH(J) = 0.
      FH(K) = 0.
  600 CONTINUE
      FH(3*N+1) = +1.
      FH(3*N+2) = 0.
      FH(3*N+3) = 0.
      IF (NOPT)4045,4050,4045
 4045 PRINT 4055
 4055 FORMAT (1H1,*(U) - OMEGABSQ (R) *)
      CALL MPRINT (UWR,N3,N3,NDIM)
      PRINT 4060
 4060 FORMAT (*1  MATRIX  (I)*)
      PRINT 4065,(AI(I),I=1,N)
 4065 FORMAT (5E20.8)
      PRINT 4070
 4070 FORMAT (*0  MATRIX  QHATB(I)*)
```

```
      PRINT 4065,(GRH(I),I=1,N)
      PRINT 4075
 4075 FORMAT (*0 MATRIX  (F)*)
      PRINT 4065,(FH(I),I=1,N3)
 4050 CONTINUE
      CALL MATINV (UWR,IROW,ICOL,N3,NDIM,SMLST)
      DO 605 I=1,N3
      APH(I) = 0.
      DO 610 J=1,N3
      APH(I) = APH(I)+UWR(I,J)*FH(J)
  610 CONTINUE
  605 CONTINUE
      IF (NOPT)4080,4085,4080
 4080 PRINT 4090
 4090 FORMAT (*0 APHAT(I) *)
      PRINT 4065, (APH(I),I=1,N3)
      PRINT 4065, (ALN(I),I=1,N)
      PRINT 4065, (X1(I),I=1,N)
      PRINT 4065, (X2(I),I=1,N)
      PRINT 4065, (X0B(I),I=1,N)
      PRINT 4065, (X1B(I),I=1,N)
      PRINT 5040
 5040 FORMAT (1H1,*(U) - OMEGABSQ (R), INVERSE *)
      CALL MPRINT (UWR,N3,N3,NDIM)
 4085 CONTINUE
      B0PH = APH(3*N+1)
      B1PH = APH(3*N+2)
      B2PH = APH(3*N+3)
      DO 225 I=1,N
      J = N+I
      CN1(I) = B2PH*X1B(I)+B1PH*X0B(I)-ALN(I)*APH(J)+ENU*APH(I)
      CN2(I) = APH(I)+ENU*(B2PH*X1B(I)+B1PH*X0B(I)-ALN(I)*APH(J))
      CN3(I) = ENU*ALN(I)*APH(I)-ALN(I)**2*APH(J)
  225 CONTINUE
      DO 230 I=1,N
      DO 230 J=1,N
      DO 230 K=1,N
      IF ( I-J+K )805,800,805
  800 TERM1 = 0.
      TERM5 = 0.
      TERM2 = (-1+(-1)**(J+K-I))/(ALN(J)+ALN(K)-ALN(I))
      TERM3 = (-1+(-1)**(I+J-K))/(ALN(I)+ALN(J)-ALN(K))
      GO TO 830
  805 IF ( J+K-I )815,810,815
  810 TERM2 = 0.
      TERM1 = (-1+(-1)**(I-J+K))/(ALN(I)-ALN(J)+ALN(K))
      TERM3 = (-1+(-1)**(I+J-K))/(ALN(I)+ALN(J)-ALN(K))
      TERM5 = (1.-(-1)**(J-I-K))/(ALN(J)-ALN(I)-ALN(K))
      GO TO 830
  815 IF ( I+J-K )825,820,825
  820 TERM3 = 0.
      TERM1 = (-1+(-1)**(I-J+K))/(ALN(I)-ALN(J)+ALN(K))
      TERM2 = (-1+(-1)**(J+K-I))/(ALN(J)+ALN(K)-ALN(I))
      TERM5 = (1.-(-1)**(J-I-K))/(ALN(J)-ALN(I)-ALN(K))
      GO TO 830
  825 TERM1 = (-1+(-1)**(I-J+K))/(ALN(I)-ALN(J)+ALN(K))
      TERM2 = (-1+(-1)**(J+K-I))/(ALN(J)+ALN(K)-ALN(I))
      TERM3 = (-1+(-1)**(I+J-K))/(ALN(I)+ALN(J)-ALN(K))
      TERM5 = (1.-(-1)**(J-I-K))/(ALN(J)-ALN(I)-ALN(K))
  830 TERM4 = (+1-(-1)**(I+J+K))/(ALN(I)+ALN(J)+ALN(K))
      DNNN(I,J,K) = -(.25/AONSQ)*(TERM1+TERM2+TERM3+TERM4)
      ENNN(I,J,K) = (.25/AONSQ)*(TERM4+TERM5-TERM3-TERM2)
```

```
  230 CONTINUE                                                          SL056300
      DO 310 J=1,N                                                      SL056400
      DO 310 K=1,N                                                      SL056500
      TERM1 = (1.-(-1)**(J+K))/(ALN(J)*ALN(K))                          SL056600
      IF ( J-K )840,835,840                                             SL056700
  835 TERM2 = 0.                                                        SL056800
      GO TO 845                                                         SL056900
  840 TERM2 = (1.-(-1)**(J-K))/(ALN(J)-ALN(K))                          SL057000
  845 EONN(J,K) = (.25/AONSQ) *(2.*(TERM1+TERM2))                       SL057100
  310 CONTINUE                                                          SL057200
      M = 10                                                            SL057300
      DO 235 I=1,N                                                      SL057400
      DO 235 J=1,N                                                      SL057500
      SUM = 0.                                                          SL057600
      DO 240 K=1,N                                                      SL057700
      SUM = SUM+ (ALN(J)**2*CN1(K)*DNNN(K,I,J)-ALN(J)*CN3(K)*ENNN(K,I,J)SL057800
     1 +M*M*CN2(K)*DNNN(K,I,J))                                         SL057900
  240 CONTINUE                                                          SL058000
      TPNN(I,J) = (SUM-ALN(J)*B2PH*EONN(I,J))*AMNK(J,JTER)              SL058100
  235 CONTINUE                                                          SL058200
      IF (NOPT)4095,5000,4095                                           SL058300
 4095 PRINT 5005                                                        SL058400
 5005 FORMAT (1H1,12X,*N1(I)*,15X,*N2(I)*,15X,*N3(I)*)                  SL058500
      PRINT 5010, (I,CN1(I),CN2(I),CN3(I),I=1,N)                        SL058600
 5010 FORMAT (I5,3E20.8)                                                SL058700
      PRINT 5035                                                        SL058800
 5035 FORMAT (*1 TP(I,J) *)                                             SL058900
      CALL MPRINT (TPNN,N,N,N)                                          SL059000
 5000 SUM1 = SUM2 = 0.                                                  SL059100
      DO 245 I=1,N                                                      SL059200
      SUM3 = SUM4 = 0.                                                  SL059300
      DO 250 J=1,N                                                      SL059400
      SUM3 = SUM3+AMNK(J,JTER)*BMMNN0(JTER,I,J)                         SL059500
      SUM4 = SUM4+AMNK(J,JTER)*BMMNNK(JTER,I,J)                         SL059600
  250 CONTINUE                                                          SL059700
      SUM1 = SUM1+AMNK(I,JTER)**2+AMNK(I,JTER)*SUM3                     SL059800
      SUM2 = SUM2+AMNK(I,JTER)**2+AMNK(I,JTER)*SUM4                     SL059900
  245 CONTINUE                                                          SL060000
      BMBAR = WKBSQ*SUM1                                                SL060100
      BKBAR = WKSQ*SUM2                                                 SL060200
      BTBAR = 0.                                                        SL060300
      DO 255 I=1,N                                                      SL060400
      DO 255 J=1,N                                                      SL060500
      BTBAR = BTBAR+TPNN(I,J)*AMNK(I,JTER)                              SL060600
  255 CONTINUE                                                          SL060700
      ABAR = (4.*BKBAR)/BMBAR                                           SL060800
      QBAR = (2.*BTBAR)/BMBAR                                           SL060900
      X0AQ = ABSF((1.-ABAR)/QBAR)                                       SL061000
      X0STR = X0AQ*((WRAD**2*A)/G)                                      SL061100
      WSTR = W/(2.*WCPS(JTER))                                          SL061200
      PRINT 4040, W,WKB,BKBAR,BTBAR,BMBAR,ABAR,QBAR,X0AQ,X0STR,WSTR     SL061300
 4040 FORMAT (1H0,2F10.2,8E12.3)                                        SL061400
      GO TO 1000                                                        SL061600
      END                                                               SL061700
      FUNCTION DKN(KD,ND)                                               DK000100
      COMMON PI,H/COM4/ALN(10)                                          DK000200
      IF (KD)10,15,10                                                   DK000300
   15 DKN = (1.0-COSF(ALN(ND)*H))/(H*ALN(ND))                           DK000400
      GO TO 20                                                          DK000500
   10 DKN = ((2.0*ALN(ND))/(H*(ALN(ND)**2-((KD*PI)/H)**2)))*(1.0-       DK000600
     1  (-1)**KD*COSF(ALN(ND)*H))                                       DK000700
   20 RETURN                                                            DK000800
```

```
      END                                                               DK000900
      FUNCTION DKNB(KDB,NPD)                                            DK000100
      COMMON PI,H/COM3/AONSQ/COM5/ALNP(10)                              DK000200
      IF (KDB)10,15,10                                                  DK000300
   15 DKNB = (1.0-COSF(ALNP(NPD)*H))/(AONSQ*ALNP(NPD))                  DK000400
      GO TO 20                                                          DK000500
   10 DKNB = (ALNP(NPD)/AONSQ)*((1.0-(-1)**KDB*CCSF(ALNP(NPD)*H))/      DK000600
     1  (ALNP(NPD)**2-((KDB*PI)/H)**2))                                 DK000700
   20 RETURN                                                            DK000800
      END                                                               DK000900
      FUNCTION RMK (MR,KR)                                              RM000100
      COMMON PI,H/COM1/A,A0,CSCOW                                       RM000200
      DIMENSION T(1000)                                                 RM000300
      XIK = SQRTF(ABSF(((KR*PI*A0)/(H*A))**2-CSCOW**2))                 RM000400
      IF (KR-(CSCOW*H)/PI)10,10,15                                      RM000500
   10 CALL BES (MR,XIK,0,RJM,T)                                         RM000600
      CALL BES (MR+1,XIK,0,RJM1,T)                                      RM000700
      RMK = RJM/(+MR*RJM-XIK*RJM1)                                      RM000800
      GO TO 20                                                          RM000900
   15 CALL BES (MR,XIK,1,RIM,T)                                         RM001000
      CALL BES (MR+1,XIK,1,RIM1,T)                                      RM001100
      RMK = RIM/(+MR*RIM+XIK*RIM1)                                      RM001200
   20 RETURN                                                            RM001300
      END                                                               RM001400
      FUNCTION EMKJ (ME,KE,JE)                                          EM000100
      COMMON PI,H/COM1/A,A0,CSCOW/COM2/UMJ( 5)                          EM000200
      DIMENSION T(1000)                                                 EM000300
      CALL BES (ME,UMJ(JE),0,EJM,T)                                     EM000400
      ALFSQ = 0.5*(1.0-(ME**2/UMJ(JE)**2))*EJM**2                       EM000500
      XIK = SQRTF(ABSF(((KE*PI*A0)/(H*A))**2-CSCOW**2))                 EM000600
      IF (KE-(CSCOW*H)/PI)10,15,15                                      EM000700
   10 EMKJ = -EJM/((XIK**2-UMJ(JE)**2)*ALFSQ)                           EM000800
      GO TO 20                                                          EM000900
   15 EMKJ = +EJM/((XIK**2+UMJ(JE)**2)*ALFSQ)                           EM001000
   20 RETURN                                                            EM001100
      END                                                               EM001200
      SUBROUTINE BES(NO,X,KODE,RESULT,T)                                BES    1
C3 UCSD BES       BESSEL FUNCTION                                       F-63
C     C3 UCSD BES                                                          F 63
       DIMENSION T(1000)                                                BES    2
 107  FORMAT(55H1NEGATIVE ORDER NOT ACCEPTED IN BESSEL FUNCTION ROUTINE)BES    3
      KLAM=1                                                            BES    4
      KO=NO+1                                                           BES    5
      IF(X) 6,1,6                                                       BES    6
 1    IF(NO) 4,2,3                                                      BES    7
 2    T(KO)=1.0                                                         BES    8
      RESULT=1.0                                                        BES    9
      RETURN                                                            BES   10
 4    IF(KO) 5,10,3                                                     BES   11
 3    RESULT=0                                                          BES   12
      RETURN                                                            BES   13
 10   RESULT=9.999999999E200                                            BES   14
      RETURN                                                            BES   15
 5     PRINT 107                                                        BES   16
      STOP1                                                             BES   17
 6    IF(NO) 5,7,7                                                      BES   18
 7    IF(KODE) 8,9,8                                                    BES   19
 8    KLAM=KLAM+1                                                       BES   20
 9    JO=2*IFIX(X)                                                      BES   21
      MO=NO                                                             BES   22
      IF(MO-JO) 11,12,12                                                BES   23
 11   MO=JO                                                             BES   24
```

```
 12    MO=MO+11                                                          BES
       T(MO)=0.                                                          BES
       LUB=MO-1                                                          BES
       T(LUB)=1.0E-250                                                   BES
       GO TO (23,51),KLAM                                                BES
 23    F=2*LUB                                                           BES
       MO=MO-3                                                           BES
       I2=MO                                                             BES
 24    F=F-2.                                                            BES
       T(I2+1)=F/X*T(I2+2)-T(I2+3)                                       BES
       IF(I2)25,26,25                                                    BES
 25    I2=I2-1                                                           BES
       GO TO 24                                                          BES
 26    SUM=T(1)                                                          BES
       DO40 J=3,MO,2                                                     BES
 40    SUM=SUM+2.*T(J)                                                   BES
       F=1./SUM                                                          BES
       DO 50 J= 1,KO                                                     BES
 50    T(J)=T(J)*F                                                       BES
       RESULT=T(KO)                                                      BES
       RETURN                                                            BES
 51    F=2*LUB-2                                                         BES
       MO=MO-3                                                           BES
       I2=MO                                                             BES
 511   T(I2+1)=F/X*T(I2+2)+T(I2+3)                                       BES
       IF(I2)52,53,52                                                    BES
 52    I2=I2-1                                                           BES
       F=F-2.                                                            BES
       GO TO 511                                                         BES
 53    SUM=T(1)                                                          BES
       DO 70 J=2,MO                                                      BES
 70    SUM=SUM+2.*T(J)                                                   BES
       F=1./SUM*EXPF(X)                                                  BES
       DO 80 J=1,KO                                                      BES
 80    T(J)=T(J)*F                                                       BES
       RESULT=T(KO)                                                      BES
       RETURN                                                            BES
       END                                                               BES
       SUBROUTINE MATINV ( A , IROW , ICOL , N , NDIM , SMLST )          MA
       DIMENSION A ( 1 ) , IROW ( 1 ) , ICOL ( 1 )                       MA
C      709-16065                                                         MA
C      709-16065  SUBROUTINE MATINV - MATRIX INVERSION ROUTINE           MA
C                                                                        MA
C          A = ARRAY NAME OF MATRIX                                      MA
C       IROW = DIMENSIONED AT N+1 OR GREATER                             MA
C       ICOL = DIMENSIONED AT N OR GREATER                               MA
C         N  = NUMBER OF EQUATIONS                                       MA
C       NDIM = VALUE OF I IN DIMENSION A(I,J) , I AND J MAY DIFFER       MA
C      SMLST = SMALLEST LEADING ELEMENT ALLOWED BEFORE CALLING THE       MA
C              SYSTEM SINGULAR , USUALLY = 1.0 E-04 OR 1.0 E-05          MA
C                                                                        MA
       NP1 = N + 1                                                       MA
       DO  100 I = 1, N                                                  MA
       ICOL ( I ) = I                                                    MA
   100 IROW ( I ) = I                                                    MA
       DO  240 ITER = 1, N                                               MA
       MAXR = ITER                                                       MA
       MAXC = 1                                                          MA
       TEMP = ABSF ( A ( MAXR ) )                                        MA
       LIMITC = NP1 - ITER                                               MA
       DO  120 I = ITER, N                                               MA
       DO  120 J = 1, LIMITC                                             MA
```

```
      IJ = ( J - 1 ) * NDIM + I
      IF ( TEMP - ( ABSF ( A ( IJ ) ) ) )  110, 120, 120
C
  110 MAXR = I
      MAXC = J
      TEMP = ABSF ( A ( IJ ) )
  120 CONTINUE
      IF ( TEMP - SMLST )  130, 130, 140
C
  130 IROW ( NP1 ) = ITER
      PRINT 1, ITER , SMLST
      RETURN
C
  140 IF ( MAXR - ITER )  150, 170, 150
C
  150 DO  160 J = 1, N
      MAXRJ = ( J - 1 ) * NDIM + MAXR
      ITJ = ( J - 1 ) * NDIM + ITER
      TEMP = A ( MAXRJ )
      A ( MAXRJ ) = A ( ITJ )
  160 A ( ITJ ) = TEMP
      ITEMP = IROW ( MAXR )
      IROW ( MAXR ) = IROW ( ITER )
      IROW ( ITER ) = ITEMP
  170 IF ( MAXC - 1 )  180, 200, 180
C
  180 DO  190 I = 1, N
      IMAXC = ( MAXC - 1 ) * NDIM + I
      TEMP = A ( I )
      A ( I ) = A ( IMAXC )
  190 A ( IMAXC ) = TEMP
      ITEMP = ICOL ( MAXC )
      ICOL ( MAXC ) = ICOL ( 1 )
      ICOL ( 1 ) = ITEMP
  200 TEMP = A ( ITER )
      ITEMP = ICOL ( 1 )
      DO  210 J = 2, N
      ITJM1 = ( J - 2 ) * NDIM + ITER
      ITJ = ( J - 1 ) * NDIM + ITER
      A ( ITJM1 ) = A ( ITJ ) / TEMP
  210 ICOL ( J - 1 ) = ICOL ( J )
      ITN = ( N - 1 ) * NDIM + ITER
      A ( ITN ) = 1 . 0 / TEMP
      ICOL ( N ) = ITEMP
      DO  240 I = 1, N
      IF ( I - ITER )  220, 240, 220
C
  220 TEMP = A ( I )
      DO  230 J = 2, N
      IJM1 = ( J - 2 ) * NDIM + I
      IJ = ( J - 1 ) * NDIM + I
      ITJM1 = ( J - 2 ) * NDIM + ITER
      A ( IJM1 ) = A ( IJ ) - A ( ITJM1 ) * TEMP
  230 CONTINUE
      IN = ( N - 1 ) * NDIM + I
      ITN = ( N - 1 ) * NDIM + ITER
      A ( IN ) = - ( TEMP * A ( ITN ) )
  240 CONTINUE
      DO  290 I = 1, N
      DO  250 J = I, N
      IF ( IROW ( J ) - I )  250, 260, 250
C
```

```
  250 CONTINUE
  260 IF ( I - J )  270, 290, 270
C
  270 DO  280 L = 1, N
      LI = ( I - 1 ) * NDIM + L
      LJ = ( J - 1 ) * NDIM + L
      TEMP = A ( LI )
      A ( LI ) = A ( LJ )
  280 A ( LJ ) = TEMP
      IROW ( J ) = IROW ( I )
  290 CONTINUE
      DO  340 I = 1, N
      DO  300 J = I, N
      IF ( ICOL ( J ) - I )  300, 310, 300
C
  300 CONTINUE
  310 IF ( I - J )  320, 340, 320
C
  320 DO  330 L = 1, N
      IL = ( L - 1 ) * NDIM + I
      JL = ( L - 1 ) * NDIM + J
      TEMP = A ( IL )
      A ( IL ) = A ( JL )
  330 A ( JL ) = TEMP
      ICOL ( J ) = ICOL ( I )
  340 CONTINUE
      IROW ( NP1 ) = 0
      RETURN
C
    1 FORMAT (7H0ON THEI3,63HTH ITERATION ALL THE REMAINING TERMS WERE L
     *ESS THAN OR EQUAL TO E11.4,18H IN ABSOLUTE VALUE)
      END
      SUBROUTINE MPRINT ( A , M , N , MD )
C     MATRIX PRINT SUBROUTINE
C     THE CALL FOR THIS SUBROUTINE IS AS FOLLOWS,
C     CALL MPRINT (A,M,N,MD)
C       WHERE A IS THE MATRIX TO BE PRINTED
C             M IS THE NUMBER OF ROWS
C             N IS THE NUMBER OF COLUMNS
C            MD IS DIMENSIONED NO. OF ROWS OF MATRIX A
      DIMENSION A ( 1 ) , JT ( 6 ) , C ( 6 )
      EQUIVALENCE ( JT , C )
      N1 = N
      N2 = 6
      N3 = 6
      N4 = 1
  100 IF ( N3 - N1 )  120, 120, 110
C
  110 N2 = N1 - N3 + 6
      N3 = N1
  120 K = 0
      DO  130 I = N4, N3
      K = K + 1
  130 JT ( K ) = I
      PRINT  1, ( JT ( I ) , I = 1 , N2 )
      DO  150 I = 1, M
      K = 0
      L = MD * ( N4 - 1 ) + I
      DO  140 J = N4, N3
      K = K + 1
      C ( K ) = A ( L )
  140 L = L + MD
```

```
  150 PRINT  2, I , ( C ( K ) , K = 1 , N2 )                                   MP    3100
      IF ( N3 - N1 )  160, 170, 170                                            MP    3200
C                                                                              MP    3300
  160 N3 = N3 + 6                                                              MP    3400
      N4 = N4 + 6                                                              MP    3500
      GO TO  100                                                               MP    3600
C                                                                              MP    3700
  170 RETURN                                                                   MP    3800
C                                                                              MP    3900
    1 FORMAT   (1H , 4X, 6( 6X, 7HCOLUMN   1I4  )   /      )                   MP    4000
    2 FORMAT   (1H  1I4, X,   (6E 17.8)    )                                   MP    4100
      END                                                                      MP    4200
      SUBROUTINE NPNRMX (A, B, N, FL, INDEX, MD, NX  )                         MTRS0148
C     CALL NPNRMX,(A, B, N, FL, INDEX, MD, NX, NP  )                           MTRS0135
C     A=VECTOR TO BE NORMALIZED  B=NORMALIZED VECTOR(MAY=A)                    MTRS0136
C     N=SIZE                      FL=NORMALIZING NUMBER                        MTRS0137
C     INDEX=+ ON ENTRY, NORMALIZE ON NUMBER WHOSE INDEX IS INDEX               MTRS0138
C        =0 ON ENTRY, NORMALIZE ON LARGEST  S.M. AND SET                       MTRS0139
C                         INDEX=TO ITS INDEX.                                  MTRS0140
C        =- ON ENTRY, NORMALIZE ON FL.                                         MTRS0141
C     MD=SINGLE PRECISION DIMENSIONED NUMBER OF ROWS OF A AND B                MTRS0142
C     NX=1, VECTOR REAL                                                        MTRS0143
C       =2,VECTOR COMPLEX                                                      MTRS0144
C     NP=1, SINGLE PRECISION                                                   MTRS0145
      DIMENSION A(1), B(1), FL(1), D(1), C(1)                                  MTRS0150
      N1 = 2                                                                   MTRS0152
      N2=N                                                                     MTRS0153
      N4=MD                                                                    MTRS0154
      IF ( INDEX) 32, 7,38                                                     MTRS0155
    7 GOTO (11,8),NX                                                           MTRS0156
    8 FL= (A(1)**2+A(N4+1)**2)                                                 MTRS0157
      INDEX=1                                                                  MTRS0158
      DO 10 K=N1,N2                                                            MTRS0259
        I=K+N4                                                                 MTRS0160
        D= (A(K)**2+A(I)**2)                                                   MTRS0161
        IF  (FL-D )  9,9,10                                                    MTRS0162
    9   FL =D                                                                  MTRS0163
        INDEX=K                                                                MTRS0164
   10   CONTINUE                                                               MTRS0165
    6 FL=A(INDEX)                                                              MTRS0166
      GOTO 18                                                                  MTRS0167
   11 FL=ABSF(A(1))                                                            MTRS0169
      INDEX=1                                                                  MTRS0170
      DO 13 K=N1,N2                                                            MTRS0171
        D=ABSF(A(K))                                                           MTRS0172
        IF (FL-D)  12,12,13                                                    MTRS0173
   12   FL=D                                                                   MTRS0174
        INDEX=K                                                                MTRS0175
   13   CONTINUE                                                               MTRS0176
   14 FL=A(INDEX)                                                              MTRS0178
   16 DO 17 I=1,N                                                              MTRS0180
   17   B(I)=A(I)/FL                                                           MTRS0181
      GOTO 30                                                                  MTRS0182
   18 I=INDEX+MD                                                               MTRS0184
      FL(2)=A(I)                                                               MTRS0185
   19 D=FL(1)**2+FL(2)**2                                                      MTRS0186
      DO 20 I=1,N                                                              MTRS0187
        K=I+MD                                                                 MTRS0188
        C=A(I)*FL(2)-A(K)*FL(1)                                                MTRS0189
        B(I)=(A(I)*FL(1)+A(K)*FL(2))/D                                         MTRS0190
   20   B(K)=-C/D                                                              MTRS0191
   30 RETURN                                                                   MTRS0211
```

```
   32 GOTO (34,36), NX                                                    MTRS0213
   34 GOTO 16                                                             MTRS0214
   36 GOTO 19                                                             MTRS0215
   38 GOTO 40                                                             MTRS0217
   40  GOTO (14,6) ,NX                                                    MTRS0219
      END                                                                 MTRS0220
      SUBROUTINE DPMLTX (A,NA,B,NB,C,M,N,K,MA,MB,MC)                      MTRS0053
C     SUBROUTINE                                                          MTRS0035
C     CALLING SEQUENCE.....                                               MTRS0037
C     CALL DPMLTX (A,NA,B,NB,C,M,N,K,MA,MB,MC)                            MTRS0038
C         A = PREMULTIPLIER        MA = DIMENSIONED NUMBER OF ROWS A      MTRS0040
C         B = POSTMULTIPLIER       MB = DIMENSIONED NUMBER OF ROWS B      MTRS0041
C         C = PRODUCT              MC = DIMENSIONED NUMBER OF ROWS C      MTRS0042
C         M = NO. ROWS IN A    = NO. ROWS IN C                            MTRS0043
C         N = NO. COLUMNS IN A = NO. ROWS IN B                            MTRS0044
C         K = NO. COLUMNS IN B = NO. COLUMNS IN C                         MTRS0045
C         NA AND NB = 1 IF A OR B, RESPECTIVELY, ARE REAL.                MTRS0046
C                   = 2 IF A OR B, RESPECTIVELY, ARE COMPLEX.             MTRS0047
C         MA,MB,MC  ARE SINGLE PRECISION DIMENSIONS,(1/2 OF ACTUAL CORE   MTRS0049
C          RESERVED FOR REAL DOUBLE PRECISION MATRICES, AND 1/4 OF        MTRS0050
C          ACTUAL CORE RESERVED FOR COMPLEX DOUBLE PRECISON MATRICES)     MTRS0051
      DIMENSION  A(1), B(1), C(1)                                         MTRS0055
      IA=MC*K                                                             MTRS0057
      IB=MA*N*NA                                                          MTRS0058
      IC=1                                                                MTRS0059
      ID=MA*NA                                                            MTRS0060
      IH=MC                                                               MTRS0061
      IJ=MC                                                               MTRS0062
      IK=M                                                                MTRS0063
      IL=1                                                                MTRS0064
      IM=0                                                                MTRS0065
      GOTO  (6,7),NA                                                      MTRS0066
    6 GOTO  (11,10),NB                                                    MTRS0067
    7 GOTO  (9,8),NB                                                      MTRS0068
    8      IC=2                                                           MTRS0069
        GOTO  10                                                          MTRS0070
    9 IH=2*IH                                                             MTRS0071
      IC=3                                                                MTRS0072
   10 IA=2*IA                                                             MTRS0073
      IJ=2*IJ                                                             MTRS074
   11 DO  30   I=1,IK,IL                                                  MTRS0075
      INC=IM                                                              MTRS0076
        DO  16   J=I,IA,IH                                                MTRS0077
      IN=INC                                                              MTRS0078
   12   C(J)=0.                                                           MTRS0081
        DO  13  L=I,IB,ID                                                 MTRS0082
          IN=IN+1                                                         MTRS0083
   13     C(J)=C(J)+A(L)*B(IN)                                            MTRS0084
   16     INC=INC+MB                                                      MTRS0091
      GOTO  (30,18,24),IC                                                 MTRS0093
   18 IE=I+MA                                                             MTRS0094
      INC=IM                                                              MTRS0095
      DO  23   J=I,IA,IJ                                                  MTRS0096
      IN=INC                                                              MTRS0097
        IF=J+MC                                                           MTRS0098
   19   DO  20   L=IE,IB,ID                                               MTRS0100
          IN=IN+1                                                         MTRS0101
          IG=IN+MB                                                        MTRS0102
          C(IF)=C(IF)+A(L)*B(IN)                                          MTRS0103
   20     C(J)=C(J)-A(L)*B(IG)                                            MTRS0104
   23   INC=INC+2*MB                                                      MTRS0112
        GOTO  30                                                          MTRS0113
```

```
24    IE=I+MC                                                                    MTRS0114
      IF=I+MA                                                                    MTRS0115
     INC=IM                                                                      MTRS0116
      DO  29  J=IE,IA,IJ                                                         MTRS0117
     IN=INC                                                                      MTRS0118
25    C(J)=0.                                                                    MTRS0120
      DO  26  L=IF,IB,ID                                                         MTRS0121
        IN=IN+1                                                                  MTRS0122
26      C(J)=C(J)+A(L)*B(IN)                                                     MTRS0123
29    INC=INC+MB                                                                 MTRS0129
30    CONTINUE                                                                   MTRS0131
     RETURN                                                                      MTRS0132
     END                                                                         MTRS0133
     SUBROUTINE SWEEPX (HTRUE, U,H, US,FL, MODE, N, MD, NC, INDEX, EP)          MTRS0233
C   COMPUTES TRUE MODE AND SWEEPS IT FROM THE MATRIX.  (REAL OR COMPLEX)         MTRS0223
C   HTRUE = TRUE MODAL COLUMNS, AS COMPUTED. U = DYNAMIC MATRIX,                 MTRS0225
C       H = SERIES OF MODIFIED MODAL COLUMNS.FL= COLUMN OF EIGENVALUES.          MTRS0226
C      US = SERIES OF MODIFIED MODAL ROWS OF U.                                  MTRS0227
C    MODE = MODE NOW BEING COMPUTED.       N = SIZE                              MTRS0228
C      MD = DIMENSIONED NUMBER OF ROWS OF U,US,H,HTRUE                           MTRS0229
C      NX = 1 IF PORBLEM IS REAL.                                                MTRS0230
C         = 2 IF PROBLEM IS COMPLEX.                                             MTRS0231
     DIMENSION H(1), US(1), U(1), HTRUE(1), FL(1), G(4)                          MTRS0235
     M=MODE-1                                                                    MTRS0237
     K1=M*NC*MD                                                                  MTRS0238
     DO   6  J=1,NC                                                              MTRS0240
       K=K1+(J-1)*MD                                                             MTRS0241
       DO   6  L=1,N                                                             MTRS0242
         K=K+1                                                                   MTRS0243
 6       HTRUE(K)=H(K)                                                           MTRS0244
     IF   ( M )  31,31,8                                                         MTRS0246
 8    DO 25   I=1,M                                                              MTRS0247
        L1=NC*MD*(MODE-I) -NC*MD                                                 MTRS0248
     GOTO ( 9,11),NC                                                             MTRS0249
 9    G=0.                                                                       MTRS0250
     DO  10   J=1,N                                                              MTRS0251
       L=L1+J                                                                    MTRS0252
       K=K1+J                                                                    MTRS0253
10     G=G+US(L)*HTRUE(K)                                                        MTRS0254
     GOTO  13                                                                    MTRS0255
11            G(1)=0.                                                            MTRS0256
              G(2)=0.                                                            MTRS0257
     DO  12  J1=1,N                                                              MTRS0258
       L=L1+J1                                                                   MTRS0259
       K=K1+J1                                                                   MTRS0260
       L2=L+MD                                                                   MTRS0261
       K2=K+MD                                                                   MTRS0262
       G(1)=G(1)+US(L)*HTRUE(K)-US(L2)*HTRUE(K2)                                 MTRS0263
12     G(2)=G(2)+US(L)*HTRUE(K2)+US(L2)*HTRUE(K)                                 MTRS0264
13   K=MODE-1                                                                    MTRS0265
     GOTO (14,19),NC                                                             MTRS0266
14   IF (ABSF(FL(K)/FL(MODE)-1.) - EP)    15,15,16                               MTRS0267
15   G=1.                                                                        MTRS0268
     GOTO  17                                                                    MTRS0269
16   G=(FL(K)-FL(MODE)) / G                                                      MTRS0270
17   DO   18   J=1,N                                                             MTRS0271
        K=K1+J                                                                   MTRS0272
        L=L1+J                                                                   MTRS0273
18      HTRUE(K)=H(L)-G(1)*HTRUE(K)                                              MTRS0274
     GOTO  25                                                                    MTRS0275
19   K=2*K                                                                       MTRS0276
     J=2*MODE                                                                    MTRS0277
```

```
      IF ( ABSF((FL(K-1)*FL(J-1)+FL(K)*FL(J))/(FL(J-1)**2+FL(J)**2)-1.) MTRS0278
     1           *EP)          20,20,22                                 MTRS0279
   20 IF ( ABSF((FL(K)*FL(J-1)-FL(K-1)*FL(J)) / (FL(J-1)**2+FL(J)**2))   MTRS028
     1       -EP)      21,21,22                                         MTRS0281
   21 G(1)=1.                                                           MTRS0282
      G(2)=0.                                                           MTRS0283
      GOTO  23                                                          MTRS0284
   22 G(3)=G(1)**2+G(2)**2                                              MTRS0285
        G(4)=(FL(K)-FL(J))*G(1)-(FL(K-1)-FL(J-1))*G(2)                  MTRS0286
        G(1)=((FL(K-1)-FL(J-1))*G(1)+(FL(K)-FL(J))*G(2)) / G(3)         MTRS0287
        G(2)= G(4) / G(3)                                               MTRS0288
   23   DO  24  J1=1,N                                                  MTRS0290
          K=K1+J1                                                       MTRS0291
          K2=K+     MD                                                  MTRS0292
          L=I1+J1                                                       MTRS0293
          L2=L+MD                                                       MTRS0294
          G(3)=HTRUE(K)                                                 MTRS0295
          HTRUE(K)= H(L)+   G(2)*HTRUE(K2)-G(1)*HTRUE(K)                MTRS0296
          HTRUE(K2)= H(L2)- G(1)*HTRUE(K2)-G(2)*G(3)                    MTRS0294
   24   CONTINUE                                                        MTRS0298
   25 CONTINUE                                                          MTRS0299
      I=0                                                               MTRS0301
      CALL NPNRMX  (HTRUE(K1+1),HTRUE(K1+1),N,C,I,MD,NC  )              MTRS0302
   31 GOTO  (26,32),NC                                                  MTRS0304
   26 DO  29  J=1,N                                                     MTRS0305
        L1=(J-1)*MD                                                     MTRS0306
        L2=K1+J                                                         MTRS0307
      DO  29  I=1,N                                                     MTRS0308
          L=L1+I                                                        MTRS0309
        IF  (I-INDEX)  28,27,28                                         MTRS0310
   27   U(L)=0.                                                         MTRS0311
        GOTO  29                                                        MTRS0312
   28   K=K1+I                                                          MTRS0313
        U(L)=U(L)-H(K)*US(L2)                                           MTRS0314
   29   CONTINUE                                                        MTRS0315
   30 RETURN                                                            MTRS0317
   32 DO 35  I=1,N                                                      MTRS0319
        L1=MD*NC*(I-1)                                                  MTRS0320
        L2=K1+I                                                         MTRS0321
          J=L2+MD                                                       MTRS0322
        DO  35  J1=1,N                                                  MTRS0323
          L=L1+J1                                                       MTRS0324
          K3=L+MD                                                       MTRS0325
          IF (J1-INDEX)    34,33,34                                     MTRS0326
   33     U(L)=0.                                                       MTRS0327
          U(K3)=0.                                                      MTRS0328
      GOTO 35                                                           MTRS0329
   34     K=K1+J1                                                       MTRS0330
          K2=K+MD                                                       MTRS0331
          U(L)=U(L)-H(K)*US(L2)+H(K2)*US(J)                             MTRS0332
          U(K3)=U(K3)-H(K2)*US(L2)-H(K)*US(J)                           MTRS0333
   35     CONTINUE                                                      MTRS0334
      GOTO 30                                                           MTRS0336
      END                                                               MTRS0337
      SUBROUTINE MITERS    (A, NTAPE, N, GUESS, NGUESS, NMODE, VECTOR,  MTRS0375
     1                      EIGVAL, NITER, NITRSP, EPSP,                MTRS0376
     2                           US, H,MAXR, NC, AITKEN, NAKSR,UTRSV)   MTRS0377
C   CALLING SEQUENCE.....                                               MTRS0341
C      A = MATRIX, DIMENSIONED (MAXR X 2*N) - REAL                      MTRS0343
C                              (MAXR X 2*N) - COMPLEX                   MTRS0344
C     N = ORDER OF MATRIX                                               MTRS0346
C     GUESS=1ST, GUESS VECTOR, DIMENSIONED (MAXR X 1) - REAL            MTRS0347
```

```
C                                             (MAXR X 2) - COMPLEX            MTRS0348
C     NGUESS=0, ROUTINE SUPPLIES GUESS VECTOR                                 MTRS0379
C           =+1, GUESS CONTAINS GUESS VECTOR                                  MTRS0350
C     NMODE=NUMBER OF EIGEN SOLUTIONS REQUESTED                               MTRS0351
C     VECTOR=EIGENVECTORS, DIMENSIONED (MAXR X NMODE) - REAL                  MTRS0352
C                                      (MAXR X 2*NMODE) - COMPLEX             MTRS0353
C     EIGVAL=EIGENVALUES               (NMODE X 1) - REAL                     MTRS0354
C                                      (NMODE*2 X 1) - COMPLEX                MTRS0355
C     NITER=NUMBER OF ITERATIONS PER MODE                                     MTRS0356
C     NITRSP = MAXIMUM NUMBER OF SINGLE PREC. ITERATIONS                      MTRS0357
C     EPSP = CONVERGENCE CRITERIA FOR SINGLE ROOTS                            MTRS0358
C     US=CHECK EIGENVECTORS,DIMENSIONED (MAXR X NMODE) - REAL                 MTRS0360
C                                       (MAXR X 2*NMODE) - COMPLEX            MTRS0361
C     H=WORKING AREA OF CORE DIMENSIONED   (MAXR X (NMODE+4) - REAL           MTRS0362
C                                          (MAXR X 2*(NMODE+4) - COMPLEX      MTRS0363
C                  WILL CONTAIN CHECK EIGENVALUES, IF REQUESTED               MTRS0364
C     MAXR = DIMENSIONED NUMBER OF ROWS                                       MTRS0367
C     NC = 1, PROBLEM REAL                                                    MTRS0368
C        = 2, PROBLEM COMPLEX                                                 MTRS0369
C     AITKEN = AITKEN CONVERGENCE CRITERIA                                    MTRS0370
C     NAKSR = NUMBER OF TIMES AITKEN APPLIED                                  MTRS0371
      DIMENSION A(1), GUESS(1), VECTOR(1), EIGVAL(1), NITER(1), US(1),        MTRS0380
     1           H(1), NAKSR(1),UTRSV(1)                                      MTRS0381
C     DEFINE PROGRAM CONSTANTS AND ZEROS.                                     MTRS0408
    8 MODE=0                                                                  MTRS0409
      AT=AITKEN**2                                                            MTRS0411
      IF ( EPSP )     12,9,12                                                 MTRS0412
    9 EPSP = .1E-08                                                           MTRS0413
   12 IF ( NGUESS )    15,13,15                                               MTRS0417
   13     J1=MAXR*(NC-1)                                                      MTRS0419
          DO  14  I=1,N                                                       MTRS0420
            K=J1+I                                                            MTRS0421
            GUESS(K)=0.                                                       MTRS0422
   14       GUESS(I)=1.                                                       MTRS0423
   15 MODE = MODE+1                                                           MTRS0425
      NAKSR(MODE)=0                                                           MTRS0426
      IGO=1                                                                   MTRS0428
      NITER(MODE)=0                                                           MTRS0429
      K1=NC*MAXR*(MODE-1)                                                     MTRS0430
      K2=K1+1                                                                 MTRS0431
      K3=NC*(MODE-1)+1                                                        MTRS0432
      K4=  NC*MAXR                                                            MTRS0433
      K5= K4*NMODE                                                            MTRS0434
      K6=K5+K4                                                                MTRS0435
C     MOVE FIRST GUESS INTO POSITION                                          MTRS0437
      DO  16  J=1,NC                                                          MTRS0438
          J1=MAXR*(J-1)                                                       MTRS0439
          DO  16  I=1,N                                                       MTRS0440
            K=K1+J1+I                                                         MTRS0441
            L=J1+I                                                            MTRS0442
   16       H(K)=GUESS(L)                                                     MTRS0443
   17 NAK=0                                                                   MTRS0445
   18 NITER(MODE)=NITER(MODE)+1                                               MTRS0446
      NAK=NAK+1                                                               MTRS0447
      INDEX=0                                                                 MTRS0448
      CALL DPMLTX (A,NC, H(K2),NC, VECTOR(K2), N,N,1, MAXR,MAXR,MAXR  )       MTRS0449
      CALL NPNRMX (VECTOR(K2), H(K2), N, EIGVAL(K3), INDEX, MAXR, NC  )       MTRS0450
C     TEST FOR SINGLE ROOT CONVERGENCE                                        MTRS0452
      DO 23 J=1,NC                                                            MTRS0453
          J1=(J-1)*MAXR                                                       MTRS0454
          K=K1+J1                                                             MTRS0455
      GOTO  (24,19,21),NAK                                                    MTRS0456
```

```
   19    L=K5+J1                                                       MTRS0457
         DO  20  I=1,N                                                 MTRS0458
            L=L+1                                                      MTRS0459
            K=K+1                                                      MTRS0460
            IF  ( ABSF(H(L)-H(K)) - EPSP)         20,20,24             MTRS0461
   20       CONTINUE                                                   MTRS0462
         GOTO  100                                                     MTRS0463
   21    DO  22  I=1,N                                                 MTRS0464
            K=K+1                                                      MTRS0465
            IF  (  ABSF(US(K)-H(K)) - EPSP  ) 22,22,24                 MTRS0466
   22    CONTINUE                                                      MTRS0467
   23 CONTINUE                                                         MTRS0466
  100 IF ACCUMULATOR OVERFLOW                      108,102             MTRS0469
  102 GOTO 56                                                          MTRS0470
C     NO CONVERGENCE, SO TEST MAXIUUM NUMBER OF ITERATIONS.            MTRS0474
   24 IF   (NITER(MODE)-NITRSP)           25,46,46                     MTRS0475
C     NOT YET EXCEEDED, SO TRY  FOR AITKENS TIME.                      MTRS0477
   25 GOTO  (40,44,31),NAK                                             MTRS0478
C     TEST FOR AITKENS CONVERGENCE.                                    MTRS0481
   31 GOTO    ( 26,36),NC                                              MTRS0482
   26 DO 28  I=1,N                                                     MTRS0484
         J=K5+I                                                        MTRS0485
         K=K1+I                                                        MTRS0486
      IF    ( US(K)-H(J) )    27,261,27                                MTRS0487
  261 IF    ( H(K)-US(K)  )    32,28,32                                MTRS0488
   27     IF   ( (ABSF(  (H(K)-US(K))/  (US(K)-H(J))   )) -AITKEN) 28,28,32MTRS0489
   28 CONTINUE                                                         MTRS0490
C    ALL VECTOR ELEMENTS OK, SO APPLY AITKENS SPEEDER-UPPER.           MTRS0492
      DO 30  I=1,N                                                     MTRS0493
         J=K5+I                                                        MTRS0494
         K=K1+I                                                        MTRS0495
      Q=(H(K)-2.*US(K)+H(J))                                           MTRS0496
      IF   (Q )   29,30,29                                             MTRS0497
   29 H(K)=H(J)- ( (US(K)-H(J))**2 / Q)                                MTRS0498
   30 CONTINUE                                                         MTRS0499
      NAKSR(MODE)=NAKSR(MODE) + 1                                      MTRS0500
      GOTO  17                                                         MTRS0501
C  CONVERGENCE TEST NOT MET, RESTORE AND TRY AGAIN.                    MTRS0503
   32 DO   33 L=1,NC                                                   MTRS0504
         J1=(L-1)*MAXR                                                 MTRS0505
         DO   33   I=1,N                                               MTRS0506
         J=K1+J1+I                                                     MTRS0507
         K=K5+J1+I                                                     MTRS0508
         H(K)=US(J)                                                    MTRS0509
   33    US(J)=H(J)                                                    MTRS0510
      NAK=2                                                            MTRS0511
      GOTO  18                                                         MTRS0512
C IF PROBLEM COMPLEX, REPEAT ALL ABOVE FOR COMPLEX ARITHMETIC.         MTRS0514
   36 DO 38  I=1,N                                                     MTRS0515
         J=K5+I                                                        MTRS0516
         K=K1+I                                                        MTRS0517
         JJ=J+MAXR                                                     MTRS0518
         KK=K+MAXR                                                     MTRS0519
         Q = (US(K)-H(J))**2 + (US(KK)-H(JJ))**2                       MTRS0520
      IF    ( Q )          37,361,37                                   MTRS0521
  361 IF ( (H(K)-US(K))**2 + (H(KK) - US(KK))**2 )    32,38,32         MTRS0522
   37   IF (  ((H(K)-US(K))**2 + (H(KK) - US(KK))**2) / Q-AT) 38,38,32 MTRS0523
   38  CONTINUE                                                        MTRS0525
      DO  39  I=1,N                                                    MTRS0527
       J=K5+I                                                          MTRS0528
      JJ=J+MAXR                                                        MTRS0529
      K=K1+I                                                           MTRS0530
```

```
      KK=K+MAXR                                                         *  MTRS0531
      Q = (H(K)-2.*US(K)+H(J))**2 + (H(KK)-2.*US(KK)+H(JJ))**2             MTRS0532
      IF  ( Q ) 35,39,35                                                   MTRS0533
   35 X=H(K)                                                               MTRS0534
      H(K)= H(J) - ( (((US(K)-H(J))**2-(US(KK)-H(JJ))**2)*(H(K)-2.*        MTRS0536
     1   US(K)+H(J))+(2.*(US(K)-H(J))*(US(KK)-H(JJ))*                      MTRS0537
     2(H(KK)-2.*US(KK)+H(JJ) ))) / Q)                                      MTRS0538
      H(KK)=H(JJ)-(((2.*(US(K)-H(J))*(US(KK)-H(JJ)))*(X-2.*                MTRS0539
     1US(K)+H(J))-((US(K)-H(J))**2-(US(KK)-H(JJ))**2)                      MTRS0540
     2*(H(KK)-2.*US(KK)+H(JJ))) / Q )                                      MTRS0541
   39 CONTINUE                                                             MTRS0542
      NAKSR(MODE) = NAKSR(MODE) + 1                                        MTRS0543
      GOTO 17                                                              MTRS0544
   40 DO 41 J=1,NC                                                         MTRS0547
      J1=MAXR*(J-1)                                                        MTRS0548
      DO  41 I=1,N                                                         MTRS0549
      K=K1+J1+I                                                            MTRS0550
      L=K5+J1+I                                                            MTRS0551
   41 H(L)=H(K)                                                            MTRS0552
      GOTO(18,56),IGO                                                      MTRS0553
   44 DO 45 J=1,NC                                                         MTRS0555
      J1=MAXR*(J-1)                                                        MTRS0556
      DO 45 I=1,N                                                          MTRS0557
      K=K1+J1+I                                                            MTRS0558
   45 US(K)=H(K)                                                           MTRS0559
      GOTO 18                                                              MTRS0560
C   MODE DID NOT CONVERGE IN NORMAL ITERATION                              MTRS0562
   46 MODE = MODE-1                                                        MTRS0564
      PRINT 94,MODE                                                        MTRS0565
      GO TO 80                                                             MTRS0566
   56 DO 58 J=1,NC                                                         MTRS0615
      J1=MAXR*(J-1)+INDEX                                                  MTRS0616
      DO 58 I=1,N                                                          MTRS0617
      K=K1+MAXR*(J-1)+I                                                    MTRS0618
      US(K)=A(J1)                                                          MTRS0619
   58 J1=J1+K4                                                             MTRS0620
      CALL SWEEPX (VECTOR,A,H,US,EIGVAL,MODE,N,MAXR,NC,INDEX,EPSP)         MTRS0621
   59 J1=(NC-1)*MAXR+INDEX                                                 MTRS0636
      GUESS(J1)=0.                                                         MTRS0637
      GUESS(INDEX)=0.                                                      MTRS0638
   62 IF  (NMODE-MODE)    70,70,15                                         MTRS0641
  108 PRINT 131                                                            MTRS0646
      MODE = MODE-1                                                        MTRS0647
      PRINT 132,MODE                                                       MTRS0648
      NMODE = MODE                                                         MTRS0649
      GO TO 70                                                             MTRS0650
   70 IF ( NTAPE ) 71,80,71
   71 CALL DPMLTX (UTRSV,NC,VECTOR,NC,US,N,N,MODE,MAXR,MAXR,MAXR)
      J = 1
      K = 1
      DO 72 I=1,MODE
      INDEX = 0
      CALL NPNRMX (US(J),US(J),N,H(K),INDEX,MAXR,NC)
      J = J+K4
   72 K = K+NC
   80 PRINT 95                                                             MTRS0666
      DO 86 I=1,MODE                                                       MTRS0667
        LITR=NITER(I)-NITRSP                                               MTRS0668
      IF ( LITR) 81,82,82                                                  MTRS0669
   81 LITR=0                                                               MTRS0670
      GOTO 85                                                              MTRS0671
   82 NITER(I)=NITRSP                                                      MTRS0672
```

```
 85 GOTO  (83,84),NC                                                    MTRS0675
 83                      PRINT 97, ( I,EIGVAL(I),NITER(I),NAKSR(I))     MTRS0676
    GOTO  86                                                            MTRS0678
 84 L=2*I-1                                                             MTRS0679
                         PRINT 96, ( I, EIGVAL(L),EIGVAL(L+1),NITER(I)  MTRS0680
   1,NAKSR(I))                                                          MTRS0681
 86 CONTINUE                                                            MTRS0682
    IF ( MODE )  92,92,88                                               MTRS0683
 88 PRINT 98                                                            MTRS0684
    L=MODE*NC                                                           MTRS0685
    CALL MPRINT  (VECTOR,N,L,MAXR)                                      MTRS0686
    IF ( NTAPE ) 92,92,90
 90 PRINT 99
    PRINT 93, (H(I),I=1,L)
    CALL MPRINT (US,N,L,MAXR)
 92 RETURN                                                              MTRS0691
 93 FORMAT (1H ,6E18.8)
 94 FORMAT (5H MODE,1I4,40H HAS NOT CONVERGED IN MAXIMUM ITERATIONS//)MTRS0693
 95 FORMAT (1H17X, 6H MODE 15X, 11H EIGENVALUE 26X,                     MTRS0694
   1        10HITERATIONS 11X, 7HAITKENS /)                             MTRS0695
 96 FORMAT (1H 1I11, 2E19.8, 1I22, 1I19 )                               MTRS0697
 97 FORMAT (1H 1I11, 9X, 1E20.8, 9X, 1I22, 1I19 )                       MTRS0698
 98 FORMAT (1H0 /   1H0 46X, 14H EIGENVECTORS ///)                      MTRS0699
 99 FORMAT (1H0 /   1H0 36H CHECK EIGENVALUES AND EIGENVECTORS )
131 FORMAT (48H1ERROR IN ITERATION SUBROUTINE....(  OVERFLOW  ))        MTRS0701
132 FORMAT (25H  CALCULATION TERMINATED.,1I6,19H MODES ARE CORRECT.)    MTRS0702
    END                                                                 MTRS0704
    END                                                                 MTRS0704
    END                                                                 MTRS0704
```